

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 08-077347

(43)Date of publication of application : 22.03.1996

(51)Int.Cl.

G06T 1/60
G06F 12/00
G06F 12/02
G06F 15/163

(21)Application number : 07-085856

(71)Applicant : TEXAS INSTR INC <TI>

(22)Date of filing : 07.03.1995

(72)Inventor : GOVE ROBERT J
KARL M GUTTAG
BALMER KEITH
READ CHRISTOPHER J
IAIN ROBERTSON
NICHOLAS ING SYMONS

(30)Priority

Priority number : 94 209123
94 209124

Priority date : 08.03.1994
08.03.1994

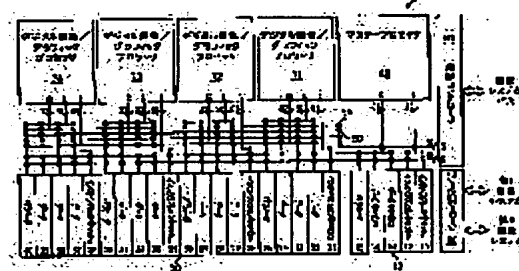
Priority country : US
US

(54) DATA PROCESSOR FOR PROCESSING IMAGE/GRAPHICS AND OPERATING METHOD THEREFOR

(57)Abstract:

PURPOSE: To improve the address control system of a memory access in a data processor for processing image/graphics.

CONSTITUTION: This data processor 100 is provided with memories 10, 20, a guide table and an address generating circuit, and a control circuit 80 receives a packet transferring request and a packet transferring parameter, which includes a start address, a guide table entry and a table pointer. The guide table includes the guide table entry and each entry includes an address value and a dimension value defining the block of an address. The address generation circuit forms the set of the blocks of addresses for a memory access corresponding to each guide table entry with the start address and a start address formed from the address value of the guide table entry.



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平8-77347

(43) 公開日 平成8年(1996) 3月22日

(51) Int.Cl. ⁶	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 T 1/60				
G 0 6 F 12/00	5 8 0	7623-5B		
12/02	5 6 0 B	7623-5B		

G 0 6 F 15/ 64 4 5 0 G

15/ 16 3 2 0 G

審査請求 未請求 請求項の数 2 書面 (全 161 頁) 最終頁に続く

(21) 出願番号 特願平7-85856

(22) 出願日 平成7年(1995) 3月7日

(31) 優先権主張番号 2 0 9 1 2 3

(32) 優先日 1994年3月8日

(33) 優先権主張国 米国 (U S)

(31) 優先権主張番号 2 0 9 1 2 4

(32) 優先日 1994年3月8日

(33) 優先権主張国 米国 (U S)

(71) 出願人 590000879

テキサス インスツルメンツ インコーポ
レイテッド

アメリカ合衆国テキサス州ダラス, ノース
セントラルエクスプレスウェイ 13500

(72) 発明者 ロバート ジェイ. ゴウブ

アメリカ合衆国テキサス州ブラノ, スカボ
ロ レーン1405

(72) 発明者 カール エム. ガタック

アメリカ合衆国テキサス州ミズーリ シテ
ィ, サウスサンデー コート 4015

(74) 代理人 弁理士 浅村 皓 (外3名)

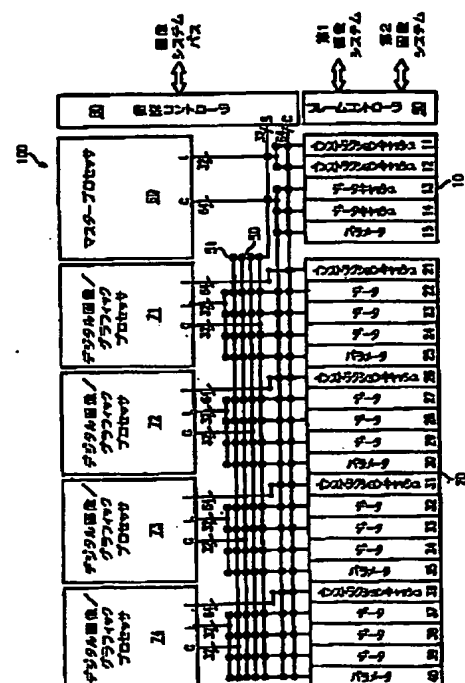
最終頁に続く

(54) 【発明の名称】 画像／グラフィックス処理用のデータ処理装置およびその操作方法

(57) 【要約】 (修正有)

【目的】 画像／グラフィックス処理用のデータ処理装置において、メモリアクセスのアドレス制御方式を改善する。

【構成】 データ処理装置100は、メモリ10、20、ガイドテーブル、及びアドレス発生回路を有し、制御回路80は、パケット転送リクエスト、及びパケット転送パラメータを受け、該パケット転送パラメータは、スタートアドレス、ガイドテーブルエントリ、及びテーブルポインタを含む。ガイドテーブルは、ガイドテーブルエントリを含み、それぞれのエントリは、アドレス値、アドレスのブロックを定義するディメンション値を含む。アドレス発生回路は、スタートアドレス、及びガイドテーブルエントリのアドレス値から形成されるスタートアドレスを有するそれぞれのガイドテーブルエントリに対応するメモリアクセス用のアドレスのブロックの集合を形成する。



【特許請求の範囲】

【請求項1】複数の対応するアドレスにデータを記憶するメモリと、

パケット転送リクエストと、パケット転送パラメータとを受ける動作をする制御回路であって、該パケット転送パラメータが、スタートアドレスと、ある数のガイドテーブルエントリと、テーブルポインタとを含む、前記制御回路と、

前記数のガイドテーブルエントリを含むガイドテーブルであって、それぞれのガイドテーブルエントリが、アドレス値およびアドレスのブロックを定義するディメンション値を含み、前記テーブルポインタが最初に前記ガイドテーブル内の第1ガイドテーブルエントリをポイントする、前記ガイドテーブルと、

前記制御回路に結合せしめられたアドレス発生回路であって、該アドレス発生回路が、前記パケット転送パラメータから、前記スタートアドレスと、前記数のガイドテーブルエントリと、前記テーブルポインタとを受ける動作をし、前記アドレス発生回路が、前記ガイドテーブルエントリに対応するメモリアクセス用のアドレスのブロックの集合を、

前記スタートアドレスと、前記テーブルポインタによりポイントされた前記ガイドテーブルエントリの前記アドレス値と、の所定の組合せから、ブロックスタートアドレスであって、該第1ブロックスタートアドレスが前記スタートアドレスである、前記ブロックスタートアドレスを形成し、

該ブロックスタートアドレスと、前記テーブルポインタによりポイントされた前記ガイドテーブルエントリの前記ディメンション値と、からアドレスのブロックを形成し、

アドレスの該ブロックにメモリアクセスを行い、もしガイドテーブルエントリの前記数よりも少ないメモリアクセスしか行われ終わっていなければ、前記メモリアクセスに続いて、前記ガイドテーブルの次のエントリをポイントするように前記テーブルポインタを更新する、ことによって形成する、前記アドレス発生回路と、を含む、データ処理装置。

【請求項2】複数の位置に情報を記憶するステップと、パケットリクエストを受けてスタートアドレスおよび制御信号を供給するステップと、

該スタートアドレスと、制御信号と、パッチ値と、に回答してパッチアドレスを発生するステップと、

パッチ情報をデコードしてディメンション値およびパッチ値を供給するステップと、

前記スタートアドレスと、前記ディメンション値と、前記パッチ値とに回答してパッチアドレスを発生するステップと、を含む、データ処理装置を操作する方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、デジタルデータ処理の技術分野にあり、より詳細には特にデジタル画像／グラフィック処理のようなデジタルデータ処理を行うマイクロプロセッサ回路アーキテクチャおよび方法に関する。

【0002】

【従来技術】本発明の実施例は用途が多く、そのうちのいくつかは説明のための背景として本明細書に説明したコンピュータグラフィックの分野に関する。ビットマップグラフィックとして知られているコンピュータグラフィックの分野では、コンピュータメモリは、画像のうちの個々の画素すなわちピクセルのデータを画像内のそのピクセルの位置に対応するメモリ位置に記憶するようになっている。この画像はディスプレイすべき画像、または操作し、記憶し、ディスプレイし、または再送信する取り込み画像とすることができる。ビットマップコンピュータグラフィックの技術分野は、ダイナミックランダムアクセスメモリ（DRAM）のコストが低下し、かつ記憶容量が増加したこと、更にマイクロプロセッサのコストが低下し、処理パワーが増大したことにより、大幅に有利となった。これらの部品のコストおよび性能が有利に変わったことにより、より大規模で、かつより複雑なコンピュータ画像システムを経済的に実現することが可能となった。

【0003】

【発明が解決しようとする課題】ビットマップグラフィックの分野は、画像データ操作に使用される処理タイプの変遷において、数回の段階を経験している。まず最初に、ビットマップグラフィックをサポートするコンピュータシステムは、すべてのビットマップ演算のためにシステムプロセッサを使用していた。このタイプのシステムには数種の欠点がある。第1の欠点として、コンピュータシステムのプロセッサは、ビットマップグラフィックの取り扱いのために特に設計されたものではないことが挙げられる。汎用計算に極めて妥当なデザイン選択は、ビットマップグラフィックシステムには不適当である。従って、ある種のルーチンググラフィックのタスクは低速でしか作動しない。更にビットマップグラフィックの画像操作のために必要な処理は、他の演算も低速にするほど、システムのプロセッサの計算容量に負荷をかけることが、すぐに判った。

【0004】ビットマップグラフィック処理の革新における次の段階は、専用ハードウェアのグラフィックコントローラにあった。これらデバイスは、システムプロセッサの制御により、簡単な図形、例えばライン、楕円および円を描画できる。これらデバイスのうちで、ピクセルブロック転送（Pixel Block Transfer）を行うこともできるデバイスもある。ピクセルブロック転送とは、メモリのある部分から他の部分へ画像データをメモリ上で移動する操作のことである。ピクセルブロック転送は、ディスプレイされていないメモリからビットマップディスプレイ

メモリに転送することにより、標準的画素、例えば特定タイプのフォントのアルファニューメリックキャラクタをディスプレイ内に入れるのに有効である。最も頻繁に使用されるグラフィック機能のうちのいくつかを行うためのビルトインアルゴリズムは、システム性能を改善する一つの方法を提供するものである。グラフィックコンピュータシステムは、かかるハードウェアのグラフィックコントローラでほとんど実現されていないような他の機能を有利に含むこともできる。これらの他の機能は、システムプロセッサによりソフトウェアで実現できる。これらのハードウェアグラフィックコントローラでは、一般にシステムプロセッサのビットマップメモリへのごく限られたアクセスを可能とし、これによりシステムソフトウェアがハードウェアのグラフィックコントローラの機能の固定された組の機能を拡張できる程度が制限されている。

【0005】グラフィックシステムプロセッサは、ビットマップグラフィック処理の革新において、別の進歩に寄与した。グラフィックシステムプロセッサとはマイクロプロセッサの属性のすべてを有し、更にビットマップグラフィックのための特殊機能を含むプログラマブルデバイスである。テキサスインスツルメンツ社によって製造されたTMS34010およびTMS34020グラフィックシステムプロセッサは、このクラスのデバイスの代表例である。これらグラフィックシステムプロセッサは、マイクロプロセッサと同様に、記憶されたプログラムに応答するもので、演算論理ユニット、レジスタファイル内のデータ記憶装置ならびにプログラムフローおよび外部データメモリの制御により、データを操作できる能力を有する。更にこれらデバイスは適当なプログラム制御で作動する特殊グラフィックオペレーション用ハードウェアを含む。これらシステムプロセッサのインストラクションセット内の付加インストラクションが、この特殊グラフィック用ハードウェアを制御するようになっている。これらインストラクションおよびこれらをサポートするハードウェアは多くのコンテキストで有効なベースレベルのグラフィック機能を実行するように選択されている。従って特定の問題のために選択されたアルゴリズムを用いることにより、多くの異なるグラフィックアプリケーションに対してプログラムできる。これにより、ハードウェア用コントローラからプログラムマイクロプロセッサに変えることによって得られる有効性の増大と同様に有効性が増す。かかるグラフィックシステムプロセッサは、マイクロプロセッサと同様にプログラム可能なデバイスであるので、これらはスタンドアロングラフィックプロセッサ、システムプロセッサにスレーブ制御されるグラフィックコプロセッサ、または密に結合されたグラフィックコントローラとして作動できる。

【0006】いくつかの分野では、経済的に実現できる

ように、よりコスト的に効果的であり、パワーのあるグラフィックオペレーションを利用することが好ましい。これらの例としては、ビデオ会議、フル動画ビデオによるマルチメディア計算、高品位テレビ、カラーファクシミリ、スマートフォトコピー、画像認識システムおよびデジタル写真がある。これらの分野の各々には固有の問題がある。画像データ圧縮およびデコンプレッションの問題はこれらアプリケーションのいくつかにおいて共通のテーマとなっている。画像および特定のフルモーションのビデオに必要な送信バンド幅の値および記憶容量は大きい。許容可能な最終画質を生じさせるような効率的なビデオ圧縮およびデコンプレッションを用いない場合、これらのアプリケーションは送信バンド幅および記憶容量に関連するコストによって制限されてしまう。この分野では、画像処理機能、例えば画像認識およびグラフィック機能、例えばディスプレイ制御の双方をサポートできる単一システムに対するニーズもある。

【0007】

【課題を解決するための手段】本発明は、メモリアクセスのアドレスの制御様式に関する。本発明のデータ処理装置は、メモリと、制御回路と、ガイドテーブルと、アドレス発生回路と、を含む。該制御回路は、パケット転送リクエストと、パケット転送パラメータとを受ける。該パケット転送パラメータは、スタートアドレスと、ある数のガイドテーブルエントリと、テーブルポインタとを含む。前記ガイドテーブルは、ガイドテーブルエントリを含み、それぞれのガイドテーブルエントリは、アドレス値およびアドレスのブロックを定義するディメンション値を含む。前記テーブルポインタは、最初に前記ガイドテーブル内の第1ガイドテーブルエントリをポイントする。前記アドレス発生回路は、それぞれのガイドテーブルエントリに対応するメモリアクセス用のアドレスのブロックの集合を形成する。前記アドレス発生回路は、前記スタートアドレスと、前記テーブルポインタによりポイントされた前記ガイドテーブルエントリの前記アドレス値と、の所定の組合せから、ブロックスタートアドレスを形成する。第1ブロックスタートアドレスは、前記スタートアドレスである。前記アドレス発生回路は、前記ブロックスタートアドレスと、前記テーブルポインタによりポイントされた前記ガイドテーブルエントリの前記ディメンション値と、からアドレスのブロックを形成する。前記アドレス発生回路は、前記ブロックスタートアドレスと、前記ガイドテーブルエントリの前記ディメンション値と、からアドレスのブロックを形成する。前記データ処理装置は、アドレスの該ブロックにメモリアクセスを行う。該メモリアクセスに続いて、前記アドレス発生回路は、前記ガイドテーブルの次のエントリをポイントするように前記テーブルポインタを更新する。

【0008】前記アドレス発生回路は、随意選択的に、

前記アドレス値を前記前のブロックスタートアドレスに加算することにより、スタートアドレスと、ガイドテーブルエントリのアドレス値と、の前記所定の組合せを形成しうる。これは、デルタガイドドアドレッシングとして公知である。前記アドレス発生回路は、随意選択的に、前記ガイドテーブル値を前記スタートアドレスに加算することにより、スタートアドレスと、ガイドテーブルエントリのアドレス値と、の前記所定の組合せを形成しうる。これは、オフセットガイドドアドレッシングとして公知である。

【0009】それぞれのガイドテーブルエントリの前記ディメンション値は、画素のアレイの水平ディメンション値および垂直ディメンション値を含む。前記メモリアクセスは、アドレスの前記ブロックからのメモリアクセスでありうる。前記メモリアクセスは、アドレスの前記ブロックへのメモリアクセスのものでありうる。実施例においては、メモリと、データプロセッサと、上述のメモリアクセスを行うデータ転送制御装置とは、単一半導体チップ内に構成されうる。該データ転送制御装置は、オンチップメモリと同様に、外部メモリにアクセスしうる。

【0010】

【実施例】図1は、本発明により画像およびグラフィック処理のために製造されたマルチプロセッサの集積回路を含む画像データ処理システムのブロック図である。このデータ処理システムは、ホスト処理システム1を含む。このホスト処理システム1は、図1のデータ処理システムのうちのホストシステムに対しデータ処理を行う。ホスト処理システム1にはプロセッサ、少なくとも一つの入力デバイス、長期記憶デバイス、リードオンリーメモリ、ランダムアクセスメモリおよびホストシステムバスに結合された少なくとも一つのホスト周辺機器が含まれる。ホスト処理システム1は、その処理機能により画像データ処理システムの機能を制御する。

【0011】マルチプロセッサ集積回路100は、図1の画像データ処理システムの画像演算のためのデータ操作および計算を含むデータ処理のほとんどを行う。マルチプロセッサ集積回路100は、画像システムバスに双方向に結合されており、この画像システムバスによりホスト処理システム1と通信するようになっている。図1の回路配列では、マルチプロセッサ集積回路100はホスト処理システム1と独立して作動する。しかしながらマルチプロセッサ集積回路100はホスト処理システム1に応答できる。

【0012】図1は2つの画像システムを示す。撮像デバイス3は画像入力デバイスとして働く文書スキャナ、電荷結合デバイススキャナまたはビデオカメラを示し、撮像デバイス3はこの画像を画像キャプチャコントローラ4へ送り、コントローラ4はこの画像をデジタル化し、画像をマスタースキャンフレームに変換するように働く。このフレームキャプチャプロセスは、マルチプロ

セッサ集積回路100からの信号により制御される。こうして形成された画像フレームは、ビデオランダムアクセスメモリ5に記憶される。ビデオランダムアクセスメモリ5は、マルチプロセッサ集積回路100により画像処理のためのデータ転送を可能とする画像システムバスを介してアクセスできる。

【0013】第2画像システムはビデオディスプレイをドライブする。マルチプロセッサ集積回路100はピクセルマップを介して、ディスプレイされる画像を指定するビデオランダムアクセスメモリ6と通信する。マルチプロセッサ集積回路100は画像システムバスを介し、ビデオランダムアクセスメモリに記憶された画像データを制御する。この画像に対応するデータはビデオランダムアクセスメモリ6から再コールされ、ビデオパレット7に供給される。ビデオパレット7は、この再コールされたデータを別のカラー空間に変換し、ピクセルごとのビット数等を拡張できる。この変換はルックアップテーブルによって行うことができる。ビデオパレット7はビデオディスプレイ8を駆動するための適当なビデオ信号も発生する。これらビデオ信号がアナログ信号であれば、ビデオパレット7は適当なデジタル-アナログ変換機能も含む。このビデオパレット7から出力されるビデオレベル信号は、カラー、飽和および輝度情報を含むことができる。マルチプロセッサ集積回路100は、ビデオパレット7内に記憶されたデータを制御し、データ変換プロセスおよび画像フレームのタイミングを制御する。マルチプロセッサ集積回路100は、ビデオパレット7の制御によりビデオディスプレイ画像のうちのフレーム当たりのライン長さおよびライン数、同期信号およびリトレース信号およびブランキング信号を制御できる。重要なことに、マルチプロセッサ集積回路100は、グラフィックディスプレイ情報をビデオランダムアクセスメモリ6内のどこに記憶するかを決定し、制御する。その後ビデオランダムアクセスメモリ6からの読み出しの間、マルチプロセッサ集積回路100はビデオランダムアクセスメモリ6からの読み出しシーケンス、アクセスすべきアドレスおよびビデオディスプレイ8に所望のグラフィック画像を発生するのに必要な制御情報を決定する。

【0014】ビデオディスプレイ8は、ユーザーによって見ることができているように、指定されたビデオデータを発生する。これには2つの技術が広く使用されている。第1の技術は、各ピクセルにおける色、色合い、輝度および飽和度の点でビデオデータを特定するものであり、第2の技術では、各ピクセルに対し、赤、青および緑のカラーレベルを指定する。ビデオディスプレイ8のためのビデオパレット7は、所定の技術とコンパチブルとなるように設計され、製造されている。図1は、画像システムバスに結合された付加メモリ9を示している。この追加メモリは、付加ビデオランダムアクセスメモリ、

ダイナミックランダムアクセスメモリ、スタティックランダムアクセスメモリまたはリードオンリーメモリを含むことができる。マルチプロセッサ集積回路100はその全体または一部をメモリ9に記憶されたプログラムにより制御できる。このメモリ9も、種々のタイプのグラフィック画像データを記憶できる。更にマルチプロセッサ集積回路100は、ビデオランダムアクセスメモリ、ダイナミックランダムアクセスメモリおよびスタティックランダムアクセスメモリのためのメモリインターフェース回路を含むことが好ましい。このように、ビデオランダムアクセスメモリ5または6を用いることなく、マルチプロセッサ集積回路100を使用してシステムを構築できる。

【0015】図1はトランシーバ16を示している。このトランシーバ16は、画像システムバスと通信チャンネルとの間で変換と双方向の通信とを行うものである。このトランシーバ16を用いたシステムの一例としてビデオ会議がある。図1に示された画像データ処理システムは、第1地点にいる人物のビデオ画像を形成するのに、撮像デバイス3と、画像キャプチャコントローラ4を用いる。マルチプロセッサ集積回路100はビデオ圧縮を行い、トランシーバ16および通信チャンネルを介して別の地点にある同様な画像データ処理システムへこの圧縮したビデオ信号を送信する。トランシーバ16は、通信チャンネルを介して遠隔地の画像データ処理システムからの同じように圧縮されたビデオ信号を受信する。マルチプロセッサ集積回路100はこの受信した信号をデコンプレスし、ビデオランダムアクセスメモリ6およびビデオパレット7を制御して、ビデオディスプレイ8上に対応するデコンプレスされたビデオ信号をディスプレイする。本発明は、画像データ処理システムがトランシーバ16を利用するような例のみに限定されるものでないことに留意されたい。更に双方向の通信は同じタイプの信号である必要はないことに留意されたい。例えば対話型ケーブルテレビ信号では、ケーブルシステムのヘッドが圧縮ビデオ信号を通信チャンネルを介して画像データ処理システムへ送り、画像データ処理システムは制御およびデータ信号をトランシーバ16および通信チャンネルを介してケーブルシステムヘッドへ送り返すことができる。

【0016】図1は、ホスト処理システム1を含むシステム内で具現化されたマルチプロセッサ集積回路100を示す。当業者であれば、マルチプロセッサ集積回路100を有効なシステムのうちの単なるプロセッサとしても使用できるような本発明の好ましい実施例を、開示内容から実現できよう。かかるシステムでは、マルチプロセッサ集積回路100はシステムの機能すべてを実行するようプログラムされる。このマルチプロセッサ集積回路100は画像処理に使用されるシステムで特に有効である。マルチプロセッサ集積回路100は複数の同一

プロセッサを含むことが好ましい。これらプロセッサの各々をデジタル画像／グラフィックプロセッサと称す。このような記載は、単に便宜的なものである。本発明を実施したプロセッサは、一つの集積回路または複数の集積回路上に別個に製造したプロセッサとすることができる。単一集積回路上に製造する場合、このような単一集積回路はデジタル画像／グラフィックプロセッサにより使用されるリードオンリーメモリおよびランダムアクセスメモリをオプションとして含むことができる。

【0017】図2は、マルチプロセッサ集積回路100のアーキテクチャを示す。マルチプロセッサ集積回路100は、2つのランダムアクセスメモリ10および20（その各々は複数の部分に別れている）と、クロスバー50と、マスタープロセッサ60と、デジタル画像／グラフィックプロセッサ71、72、73および74と、システムメモリへのアクセスを仲介する転送コントローラ80と、独立した第1および第2画像メモリへのアクセスを制御できるフレームコントローラ90とを含む。マイクロプロセッサ集積回路100は、高度のオペレーションパラレリズム（並行性）を提供する。これは画像処理およびグラフィックオペレーション、例えばマルチメディアの計算において有効である。これらプロセッサが有効となる画像およびグラフィック処理以外の計算アプリケーションもあるので、画像／グラフィックプロセッサとしてプロセッサ71、72、73および74を参考としたことは、単なる便宜的なものである。

【0018】マイクロプロセッサ集積回路100は2つのランダムアクセスメモリを含む。ランダムアクセスメモリ10は、主にマスタープロセッサ60の専用である。このメモリは2つのインストラクションキャッシュメモリ11および12と、2つのデータキャッシュメモリ13および14と、パラメータメモリ15を含む。これらメモリ部分は物理的には同一とすることができるが、異なるように接続し、使用することもできる。ランダムアクセスメモリ20はマスタープロセッサ60およびデジタル画像／グラフィックプロセッサ71、72、73および74の各々によってアクセスできる。各デジタル画像／グラフィックプロセッサ71、72、73および74は、5つの対応するメモリ部分を有する。これらはインストラクションキャッシュメモリ、3つのデータメモリおよび1つのパラメータメモリを含む。従ってデジタル画像／グラフィックプロセッサ71は対応するインストラクションキャッシュメモリ21、データメモリ22、23、24およびパラメータメモリ25を有し、デジタル画像／グラフィックプロセッサ72は対応するインストラクションキャッシュメモリ26と、データメモリ27、28、29およびパラメータメモリ30を有し、デジタル画像／グラフィックプロセッサ73は対応するインストラクションキャッシュメモリ31と、データメモリ32、33、34およびパラメータメモリ

35を有し、デジタル画像／グラフィックプロセッサ74は対応するインストラクションキャッシュメモリ36と、データメモリ37、38、39およびパラメータメモリ40とを有する。ランダムアクセスメモリ10の部分と同じように、これらメモリ部分は物理的に同一にできるが、異なるように接続し、使用することもできる。メモリ10および20のうちのこれらメモリ部分の各々は、例えば20Kバイトを含み、マルチプロセッサ集積回路100内の総メモリは50Kバイトとなる。

【0019】マルチプロセッサ集積回路100は、複数の独立したパラレルデータ転送を用いることにより、プロセッサとメモリとの間で高レートでのデータ転送を行うように製造されている。クロスバー50はこれらのデータ転送を可能にするものであり、各デジタル画像／グラフィックプロセッサ71、72、73および74は、サイクルごとに同時に作動できる3つのメモリポートを有する。インストラクションポート(I)は対応するインストラクションキャッシュから64ビットのデータワードをフェッチでき、ローカルデータポート(L)はデジタル画像／グラフィックプロセッサに対応するデータメモリまたはパラメータメモリから32ビットのデータワードを読み出したり、これに書き込みを行うことができる。グローバルデータポート(G)は、データメモリまたはパラメータメモリまたはランダムアクセスメモリ20のいずれかから32ビットのデータワードを読み出したり、書き込んだりできる。マスタープロセッサ60は、2つのメモリポートを含む。インストラクションポート(I)はインストラクションキャッシュ11および12のいずれかから32ビットのインストラクションワードをフェッチできる。データポート(C)はデータキャッシュ13または14、ランダムアクセスメモリ10のうちのパラメータメモリ15、またはデータメモリ、パラメータメモリ、ランダムアクセスメモリ20の任意のものから32ビットのデータワードを読み出したり、これらに書き込みできる。転送コントローラ80はデータポート(C)を介してランダムアクセスメモリ10または20の部分のいずれかにアクセスできる。従って一つのメモリサイクルで16のパラレルメモリアクセスをリクエストできる。このような多数のパラレルアクセスをサポートするように、ランダムアクセスメモリ10および20は、25のメモリに分割されている。

【0020】クロスバー50はマスタープロセッサ60、デジタル画像／グラフィックプロセッサ71、72、73および74、および転送コントローラ80とメモリ10および20との接続を制御する。クロスバー50は、列と行に配置された複数のクロスポイント51を含む。クロスポイント51の各列は、単一メモリ部分および対応するアドレスのレンジに対応する。プロセッサはこのプロセッサが出力するアドレスの最高位ビットによるメモリ部分のうちの一つへのアクセスをリクエスト

する。プロセッサにより出力されるこのアドレスは、行に沿って進む。このアドレスを有するメモリ部分に対応するクロスポイント51は、メモリ部分へのアクセスを許可または否定することによって応答する。それ以外のいずれのプロセッサも、そのときのメモリサイクルの間にそのメモリ部分へのアクセスをリクエストしない場合、クロスポイント51が行と列を結合することにより、アクセスを許可する。これによりそのメモリ部分にアドレスが供給される。このメモリ部分はそのアドレスにおけるデータアクセスを可能にすることに応答する。このデータアクセスはデータ読み出しオペレーションまたはデータ書き込みオペレーションのいずれかでよい。

【0021】2つ以上のプロセッサが同じメモリ部分へのアクセスを同時にリクエストする場合、クロスバー50はリクエスト中のプロセッサのうちの一つのアクセスしか許可しない。クロスバー50のうちの各列のうちのクロスポイント51は、優先階層に基づいて通信し、アクセスを許可する。同じランクを有するアクセスの2つのリクエストが同時に行われた場合、クロスバー50は最後に許可されたプロセッサが最も低い優先度を有するラウンドロビン法によりアクセスを許可する。リクエストにサービスする必要がある限り、各々の許可されたアクセスが続く。プロセッサはメモリサイクルごとにアドレスを変更できるので、クロスバー50はサイクルごとにプロセッサとメモリ部分との間の相互接続を変更できる。マスタープロセッサ60は、マルチプロセッサ集積回路100のための主要制御機能を実行することが好ましい。マスタープロセッサ60はハードウェアの浮動小数点計算ユニットを含む32ビットの縮小インストラクションセットコンピュータ(RISC)プロセッサであることが好ましい。RISCアーキテクチャによれば、メモリへのすべてのアクセスは、ロードおよび記憶インストラクションで実行され、ほとんどの整数および論理演算はレジスタ上で1回のサイクルで実行される。しかしながら整数および論理ユニットによって使用されるのと同じレジスタファイルを用いる場合、一般に演算を実行するには2サイクルがかかる。レジスタのスコアボードは正しいレジスタアクセスシーケンスを維持するように保証するものであり、画像処理における制御機能に対してはRISCアーキテクチャが適当である。浮動小数点計算ユニットは画像回転機能の高速計算を可能にするもので、このような機能は画像処理に重要である。

【0022】マスタープロセッサ60は、インストラクションキャッシュメモリ11またはインストラクションキャッシュメモリ12からのインストラクションワードをフェッチする。同じように、マスタープロセッサ60はデータキャッシュ13またはデータキャッシュ14のいずれかからデータをフェッチする。各メモリ部分は2Kバイトのメモリを含むので、14Kバイトのインストラクションキャッシュと4Kバイトのデータキャッシュ

があることになる。キャッシュ制御はマスタープロセッサ60の不可欠な機能であり、上記のようにマスタープロセッサ60はクロスバー50を介して他のメモリ部分にもアクセスできる。4つのデジタル画像／グラフィックプロセッサ71、72、73および74の各々は、高度にパラレルなデジタル信号プロセッサ(DSP)アーキテクチャを有する。デジタル画像／グラフィックプロセッサ71、72、73および74は3つの別個のユニット、すなわちデータユニットと、アドレスユニットと、プログラムフロー制御ユニットを用いた高度なオペレーションのパラレリズムが得られる。これら3つのユニットは、インストラクションパイプライン内の異なる命令で同時に作動する、更にこれらユニットの各々は、内部パラレリズムを含む。

【0023】デジタル画像／グラフィックプロセッサ71、72、73および74は、マルチインストラクションマルチデータモード(MIMD)で独立したインストラクションストリームを実行できる。このMIMDモードでは、各デジタル画像／グラフィックプロセッサは、対応するインストラクションキャッシュからの個々のプログラム(これは独立的でも協働的でもよい)を実行する。後者のケースでは、クロスバー50は共用メモリと共に、プロセッサ内通信を可能にする。デジタル画像／グラフィックプロセッサ71、72、73および74は、同期MIMDモードでも作動できる。同期MIMDモードでは、各デジタル画像／グラフィックプロセッサのプログラムフロー制御ユニット130は、すべての同期したプロセッサが進む準備がなされるまで、次のインストラクションのフェッチを禁止する。このような同期MIMDモードは密に結合されたオペレーションにおけるロックステップでデジタル画像／グラフィックプロセッサの別々のプログラムを実行できるようにするものである。

【0024】デジタル画像／グラフィックプロセッサ71、72、73、74は、単一インストラクションマルチデータモード(SIMD)で異なるデータに対する同じインストラクションを実行できる。このモードにおいて、4つのデジタル画像／グラフィックプロセッサのための単一インストラクションストリームは、インストラクションキャッシュメモリ21から生じる。データユニット画像／グラフィックプロセッサ71は、フェッチオペレーションおよびブランチオペレーションを制御し、クロスバー50は他のデジタル画像／グラフィックプロセッサ72、73および74へ同じインストラクションを供給する。デジタル画像／グラフィックプロセッサ71は、デジタル画像／グラフィックプロセッサ71、72、73および74のすべてに対するインストラクションフェッチを制御するので、これらデジタル画像／グラフィックプロセッサは本来SIMDモードでは同期化されるものである。

【0025】転送コントローラ80は、マルチプロセッサ集積回路100のための組み合わせられたダイレクトメモリアクセス(DMA)マシンと、メモリインターフェースであり、この転送コントローラ80は、インテリジェントにキューイングし、優先度をセットし、5つのプログラマブルプロセッサのデータリクエストおよびキャッシュミスにサービスする。マスタープロセッサ60およびデジタル画像／グラフィックプロセッサ71、72、73および74のいずれも、転送コントローラ80を介してマルチプロセッサ集積回路100の外部のメモリおよびシステムにアクセスする。データキャッシュまたはインストラクションキャッシュミスは転送コントローラ80によって自動的に取り扱われる。キャッシュサービス(S)ポートは、かかるキャッシュミスを転送コントローラ80へ伝送し、キャッシュサービスポート(S)はメモリからではなくプロセッサから情報を読み出すマスタープロセッサ60およびデジタル画像／グラフィックプロセッサ71、72、73および74は、リンクされたリストパケット転送として転送コントローラ80からのデータ転送をリクエストできる。これらリンクされたリストパケット転送は、ソースメモリアドレスと宛て先メモリアドレスとの間で多次元ブロックの情報の転送を可能にするものであり、これらアドレスはマルチプロセッサ集積回路100内にあってもよいし、またはマルチプロセッサ集積回路100の外部にあってもよい。転送コントローラ80は、内部のデータを保持するのに周期的なりフレッシュを必要とするダイナミックランダムアクセスメモリ(DRAM)用のリフレッシュコントローラも含むことが好ましい。

【0026】フレームコントローラ90はマルチプロセッサ集積回路100と、外部画像キャプチャおよびディスプレイシステムとの間のインターフェースである。このフレームコントローラ90は、キャプチャおよびディスプレイデバイスに対する制御を行い、これらデバイスとメモリとの間のデータの移動を自動的に管理する。このため、フレームコントローラ90は、2つの独立した画像システムに対して同時に制御を行う。これら画像システムは一般に画像キャプチャ(取り込み)用第1画像システムと画像ディスプレイ用の第2画像システムとからなるが、フレームコントローラ90の使用はユーザーによって制御される。これら画像システムは、通常、フレームグラバまたはフレームバッファ記憶装置のいずれかに対して使用される独立したフレームメモリを含む。フレームコントローラ90はリフレッシュおよびシフトレジスタ制御により、ビデオダイナミックランダムアクセスメモリ(VRAM)を制御するように作動することが好ましい。

【0027】マルチプロセッサ集積回路100は、大規模画像処理のために設計されている。マスタープロセッサ60は埋め込み制御を行い、デジタル画像／グラフィ

ックプロセッサ71、72、73、74の活動を調和させ、これらプロセッサの発生した結果を解釈する。デジタル画像／グラフィックプロセッサ71、72、73、74は、ピクセル解析および操作に良好に適している。ピクセルのデータが多く、情報が少ないとみなされる場合、代表的なアプリケーションではデジタル画像／グラフィックプロセッサ71、72、73、74がピクセルを良好に検査し、未加工データを情報とする。次にこの情報は、デジタル画像／グラフィックプロセッサ71、72、73、74またはマスタープロセッサ60のいずれかによって分析できる。クロスバー50はプロセッサ内通信を仲介する。更にこのクロスバー50はマスタープロセッサ集積回路100を共用メモリシステムとして実現できるようにする。このアーキテクチャでは、メッセージの通過は通信の主な形態となる必要はない。しかしながら共用メモリをメッセージが通過するようにできる。各デジタル画像／グラフィックプロセッサ、クロスバー50の対応する部分およびメモリ20の対応する部分は、この実施例内で同じ幅を有する。これにより、同じピン出力を維持しながら、モジュラー式のデジタル画像／グラフィックプロセッサおよび対応するメモリの追加または取り外しに適應できるようにすることにより、アーキテクチャをフレキシブルにできる。

【0028】ある実施例では、マルチプロセッサ集積回路100のすべての部品は、 $0.6\mu m$ の特徴的寸法を用いた相補的酸化金属半導体(CMOS)に形成された単一集積回路に配置されている。マルチプロセッサ集積回路100は、256個のピンを有するピングリッドアレイパッケージ内に適当に製造される。入出力端は、例えばTTL論理電圧にコンパチブルであり、マルチプロセッサ集積回路100は、約300万個のトランジスタを含み、50MHzのクロックレートを使用している。図3は、インタラプトイネーブルレジスタINTEN110およびインタラプトフラグレジスタINTFLG115のためのフィールド定義を示す。rと表示されたビットは、将来の使用に対して保留されており、一のついたビットは、好ましい実施例では使用されないが、別の実施例では使用できる。インタラプト(割り込み)は、左から右に優先度が決められている。各インタラプトソースは、インタラプトイネーブルレジスタINTEN110の対応するイネーブル(E)ビット内に1をセットすることによって、個々にイネーブルできる。インタラプトフラグレジスタINTFLG115のインタラプトソースビットは、右から左へ、すなわち常時イネーブルにされているエミュレーションインタラプトETRAP、XYパッチインタラプト、タスクインタラプト、パケット転送ビジーインタラプトPTB、パケット転送エラーインタラプトPTERROR、パケット転送成功インタラプトPTEND、マスタプロセッサ60メッセージインタラプトMPMSG、デジタル画像／グラフィックプ

ロセッサ71メッセージインタラプトDIGPOMSG、デジタル画像／グラフィックプロセッサ72メッセージインタラプトDIGP1MSG、デジタル画像／グラフィックプロセッサ73メッセージインタラプトDIGP2MSG、デジタル画像／グラフィックプロセッサ74メッセージインタラプトDIGP3MSGの方向に、優先度が低くなるようになっている。ビット31～28は、8個のデジタル画像／グラフィックプロセッサを含むマルチプロセッサ集積回路100の実現の際に4つの追加デジタル画像／グラフィックプロセッサからのメッセージインタラプトのために保留されている。

【0029】インタラプトイネーブルレジスタINTEN110のWビット(ビット0)はインタラプトフラグレジスタINTFLG115への書き込みを制御する。通常このビットはエミュレーションインタラプトをイネーブルするかどうかを制御する。好ましい実施例では、エミュレーションインタラプトはディスエーブルできないので、インタラプトイネーブルレジスタINTEN110内で、このインタラプトに対するイネーブルビットの必要はない。インタラプトイネーブルレジスタINTEN110のビット0は、インタラプトフラグレジスタINTFLG115の作動を変えるものである。インタラプトイネーブルレジスタINTEN110のWビットが1であると、インタラプトフラグレジスタINTFLG115へのソフトウェアの書き込みはビットを1にセットできるだけである。これらの条件では、インタラプトフラグレジスタINTFLG115のビットへの0の書き込みは全く効果はない。このようなWビットが0である場合、インタラプトフラグレジスタINTFLG115のいずれかのビットへの1の書き込みは、そのビットを0にクリアする。インタラプトフラグレジスタINTFLG115の任意のビットへの0への書き込みは全く効果がない。これにより、他のステートを乱すことなくインタラプトフラグレジスタINTFLG115内の個々のインタラプトフラグをクリアできる。各インタラプトサービスルーチンは、リターン前に対応するインタラプトフラグを適当にクリアするが、その理由は好ましい実施例ではハードウェアによってこれらフラグをクリアしていないからである。この唯一の例外として、エミュレーションインタラプトETRAPハードウェアによってクリアされるが、この理由は、このようなインタラプトが常時イネーブルされているからである。特定のインタラプトソースがソフトウェアの書き込みによってこれをクリアするのと同じように、インタラプトフラグレジスタINTFLG115内のビットを同時にセットしようとしている場合、ロジックによりこのビットはセットされる。

【0030】ETRAPインタラプトフラグ(インタラプトフラグレジスタINTFLG115のビット0)は、解析論理またはETRAPインストラクションのい

ずれからセットされる。このインタラプトはディスエーブルされないで、通常は即座にサービスされるが、インタラプトサービスはパイプラインストール条件、例えばクロスバ50を介するメモリの競合が解決されるまで待機する。ENTRAPインタラプトフラグは、インタラプトサービスがされる際にハードウェアによってクリアされるインタラプトフラグレジスタINTFLG115内の単なるインタラプトビットである。XY PATCHインタラプトフラグ（インタラプトフラグレジスタINTFLG115のビット11）は、XYアドレス指定をするのにグローバルアドレスユニット610と、ローカルアドレスユニット620との組み合わせを用いる際に、所定の条件でセットされる。XYパッチされたアドレス指定は所定の条件でインタラプトを発生できる。XYパッチされたアドレス指定のためのインストラクションワードの呼び出しは、かかるインタラプトを発生できるかどうか、更に指定されたパッチの内外でアドレス上で許可されたインタラプトを行うかどうかを示している。

【0031】マスタプロセッサ60からのコマンドを受信した際に、TASKインタラプトフラグ（インタラプトフラグレジスタINTFLG115内のビット14）がセットされる。このインタラプトによりデジタル画像／グラフィックプロセッサ71がそのTASKインタラプトベクトルをロードする。このインタラプトは、例えばマスタプロセッサ60の制御により、選択されたデジタル画像／グラフィックプロセッサ71、72、73、74をスイッチングさせることができる。キューアクティブビットが1のとき、ソフトウェアが通信用レジスタCOMM120のパケット転送ビットに1を書き込む場合、パケット転送ビジーインタラプトフラグPTB（インタラプトフラグレジスタINTFLG115のビット17）がセットされる。これにより先のパケット転送が終了したかどうかをチェックすることなくパケット転送を送ることができる。先のパケット転送がまだ待機中であれば、このインタラプトフラグはセット状態となる。これについては通信用レジスタCOMM120の説明と関連して後に詳細に述べる。

【0032】デジタル画像／グラフィックプロセッサにより送られるパケット転送を実行する間に、転送用コントローラ80がエラー条件に遭遇すれば、パケット転送エラーインタラプトフラグPERROR（インタラプトフラグレジスタINTFLG115のビット18）がセットされる。転送コントローラ80がデジタル画像／グラフィックプロセッサのリンクされたリストの終了部に遭遇するか、または終了時にリクエスト中のデジタル画像／グラフィックプロセッサをインタラプトするよう、転送コントローラ80に命令するパケット転送を完了した際に、パケット転送エンドインタラプトフラグP

TEND（インタラプトフラグレジスタINTFLG115のビット19）が転送コントローラ80によってセットされる。

【0033】マスタプロセッサ60がそのデジタル画像／グラフィックプロセッサにメッセージインタラプトを送ると、マスタプロセッサメッセージインタラプトフラグMPMSG（インタラプトフラグレジスタINTFLG115のビット20）がセット状態となる。インタラプトフラグレジスタINTFLG115のビット27～24は、デジタル画像／グラフィックプロセッサ71、72、73、74からのメッセージインタラプトをログする。デジタル画像／グラフィックプロセッサ71、72、73または74は、インタラプトフラグレジスタINTFLG115の対応するビットを介してそれ自体にメッセージを送ったり、それ自体をインタラプトできる。デジタル画像／グラフィックプロセッサ71が、デジタル画像／グラフィックプロセッサにメッセージインタラプトを送ると、デジタル画像／グラフィックプロセッサ0のメッセージインタラプトフラグDIGPMSG（インタラプトフラグレジスタINTFLG115のビット24）がセットされる。同様に、デジタル画像／グラフィックプロセッサ72が、メッセージインタラプトを送ると、デジタル画像／グラフィックプロセッサ1のメッセージインタラプトフラグDIGP1MSG（インタラプトフラグレジスタINTFLG115のビット25）がセットされる。デジタル画像／グラフィックプロセッサ73が、メッセージインタラプトを送ると、デジタル画像／グラフィックプロセッサ2のメッセージインタラプトフラグDIGP2MSG（インタラプトフラグレジスタINTFLG115のビット26）がセットされる。デジタル画像／グラフィックプロセッサ74が、メッセージインタラプトを送ると、デジタル画像／グラフィックプロセッサ3のメッセージインタラプトフラグDIGP3MSG（インタラプトフラグレジスタINTFLG115のビット27）がセットされる。先に述べたように、インタラプトフラグレジスタINTFLG115のビット31～28は、8個のデジタル画像／グラフィックプロセッサを含むマルチプロセッサ集積回路100を実現する際の4つの付加的デジタル画像／グラフィックプロセッサからのメッセージインタラプトのために保留されている。

【0034】イネーブルされたインタラプトが起きると、小ステートマシンにすることができるインタラプト疑似インストラクションユニットがインストラクションレジスターアドレスステージでパイプラインに次の疑似インストラクションの組を注入する。

【0035】

【数1】

17
 *(A14 - 16) = SR
 *(A14 + 12) = PC
 BR = *vectadd ; S, GおよびLビットは、
 vectaddの2つの最小位ビット=11
 *(A14 + 8) = IPA
 *(A14 + 4) = IPE

18

【0036】これら疑似インストラクションをそれぞれPS1、PS2、PS3、PS4およびPS5と称す。このシーケンスによってはサブルーチンIPRSからのインストラクションポインタリターンはセーブされない。インタラプトサービスルーチンがなんらかのブランチを実行する場合、インタラプトサービスルーチンによってまず最初にサブルーチンIPRSからのインストラクションポインタリターンをプッシュし、リターン前にレストアしなければならない。ベクトルフェッチは保護されているサブルーチンIPRSからのインストラクションポインタリターンを全プログラムカウンタPC701にロードすることである。これによりプログラムカウンタPC701のS、GおよびLビットがロードされるので、すべてのインタラプトベクトルの3つの最小位ビットが0にされる。この説明の例外は、リセット後にフェッチされるタスクベクトルは、ルーピングをディスエーブルするようにLビット（プログラムカウンタPC701のビット0）をセットさせなければならないことである。

【0037】インタラプトフラグレジスタINTFLG115内に示されたインタラプトのためのインタラプトサービスルーチンの開始ポイントのそれぞれのアドレスのデジタル画像／グラフィックプロセッサインタラプト

ベクトルと称す。これらアドレスはソフトウェアによって発生され、表1に示されるそれぞれのインタラプトされたデジタル画像／グラフィックプロセッサ71、72、73、74に対応するパラメータメモリ25、30、35、40にデータとしてロードされる。インタラプト疑似インストラクションPS3は、対応するパラメータメモリ25、30、35または40における表示されたアドレスに記憶された32ビットのアドレスを取り出し、これをプログラムカウンタPC701に記憶する。インタラプト疑似インストラクションユニット770はインタラプトイネーブルレジスタによってイネーブルされた最高優先度のインタラプトに基づき、対応するパラメータメモリのためにアドレスを計算する。インタラプト疑似インストラクションユニット770は各デジタル画像／グラフィックプロセッサのためのユニークなアドレスを発生するよう、通信レジスタCOMM120からのデジタル画像／グラフィックプロセッサ番号を含むように作動する。インタラプト疑似インストラクションPS4およびPS5は、インタラプトサービスルーチンへのブランチの後のディレイスロット内にあることに留意されたい。

【0038】

【表1】

10

20

30

INTFLG ビット	インタラプト名	アドレス
31	DIGP7メッセージ用に保留	0100#1FC
30	DIGP6メッセージ用に保留	0100#1F8
29	DIGP5メッセージ用に保留	0100#1F4
28	DIGP4メッセージ用に保留	0100#1F0
27	DIGP3メッセージ	0100#1EC
26	DIGP2メッセージ	0100#1E8
25	DIGP1メッセージ	0100#1E4
24	DIGP0メッセージ	0100#1E0
23	スベア	0100#1DC
22	スベア	0100#1D8
21	スベア	0100#1D4
20	マスタプロセッサメッセージ	0100#1D0
19	パケット転送成功	0100#1CC
18	パケット転送エラー	0100#1C8
17	パケット転送ビジー	0100#1C4
16	スベア	0100#1C0
15	スベア	0100#1BC
14	TASKインタラプト	0100#1B8
13	スベア	0100#1B4
12	スベア	0100#1B0
11	XYパッチング	0100#1AC
10	保留	0100#1A8
9	保留	0100#1A4
8	保留	0100#1A0
7	保留	0100#19C
6	保留	0100#198
5	保留	0100#194
4	保留	0100#190
3	保留	0100#18C
2	スベア	0100#188
1	スベア	0100#184
0	エミュレーション	0100#180

表1

【0039】各アドレスにて#は通信レジスタCOMM 120から得られたデジタル画像／グラフィックプロセッサ番号と置換される。

【0040】インタラプトサービスルーチンの最後の4つのインストラクションは、次の(32ビットデータのシフトされていないインデックス)オペレーションを含んでいなければならない。

【0041】

【数2】

SR = *(A14 ++ 4)

BR = *(A14 ++ 7)

BR = *(A14 ++ 5)

BR = *(A14 ++ 5)

【0042】これらインストラクションをそれぞれRETI1、RETI2、RETI3およびRETI4と称す。他のオペレーションは希望すればこれらオペレーションと平行にコード化できるが、これらのオペレーションのいずれもステータスレジスタ211を変えてはならない。

【0043】新しいタスクをデジタル画像／グラフィックプロセッサ上で実行すべきであり、新しいタスクを終了した後最初のステートに復帰すべき場合インタラプトステートをセーブできる。インタラプトイネーブルレジスタINTEN110上のWビットにより制御される

書き込みモードにより、セービングまたはレストアオペレーション中にインタラプトを失うことなくこれを行うことができる。これは次のインストラクションシーケンスで実行できる。まずDINTインストラクションによりインタラプトをディスエーブルし、次にインタラプトイネーブルレジスタINTEN110およびインタラプトフラグレジスタINTFLG115の双方をセーブし、インタラプトイネーブルレジスタINTEN110のWビット(ビット0)を0にセットし、16進数FFFFFFFFをインタラプトフラグレジスタINTFLG115に書き込む。イネープリングインタラプトを含むことができる新しいタスクを実行する。新しいタスクの完了後、元のタスクをリカバーし、最初にDINTインストラクションによりインタラプトをディスエーブルし、インタラプトイネーブルレジスタINTEN110のWビットを1にセットする。メモリからインタラプトフラグレジスタINTFLG115のステータスをレストアし、次にメモリからインタラプトイネーブルレジスタINTEN110のステータスをレストアする。最後にEINTインストラクションによりインタラプトをイネーブルする。

【0044】各デジタル画像／グラフィックプロセッサ71、72、73、74は、他のデジタル画像／グラフィックプロセッサおよびマスタプロセッサ60にコマン

ドワードを送信できる。レジスタA15の宛て先、グローバルアドレスユニットのうちの0値のアドレスレジスタによるレジスタ間の移動により、宛て先プロセッサへのコマンドワードの転送が開始される。このようなレジスタ間の転送は、後述するように、一つのインストラクションでデータユニット110のオペレーションおよびローカルデータポート144を介したアクセスと組み合わせることができる。このようなコマンドワードは特別コマンドワード信号に伴われてグローバルデータポート148を介してクロスバー50へ送信される。これによりマスタプロセッサ60およびデジタル画像/グラフィックプロセッサ71、72、73、74はマルチプロセッサ集積回路100の他のプロセッサとの通信が認められる。

【0045】図4は、これらコマンドワードのフィールド定義を略図で示している。好ましい実施例では、コマンドワードはグローバルデータポート148を介して送信されるデータと同一の32ビット長さを有する。各コマンドワードの最小位のビットは、コマンドワードがアドレス指定される一つ以上のプロセッサおよび他の回路を定義している。各受け入れ側回路は、これらビットがコマンドワードをその回路に向けることを表示している場合に限り、受信したコマンドワードに応答する。各コマンドワードのビット3~0はそれぞれデジタル画像/グラフィックプロセッサ74、73、72、71をそれぞれ指定している。好ましい実施例ではビット7~4は使用されず、8個のデジタル画像/グラフィックプロセッサを有するマルチプロセッサ集積回路100内で使用するために保留されている。ビット8はマルチプロセッサ60にコマンドワードをアドレス指定することを表示している。ビット9はコマンドワードを転送コントローラ80に向けることを表示しており、ビット10はコマンドワードをフレームコントローラ90に向けることを表示している。ここですべての回路はすべてのコマンドワードを他のすべての回路に送ることを許可されているわけではないことに留意されたい。例えばシステムレベルのコマンドワードはデジタル画像/グラフィックプロセッサから他のデジタル画像/グラフィックプロセッサまたはマスタプロセッサ60へ送ることはできない。マスタプロセッサ60しか、転送コントローラ80またはフレームコントローラ90へコマンドワードを送ることができない。どの回路がどのコマンドワードを他のどの回路に送ることができるかの制限について、各コマンドワードフィールドの説明と関連させて後に説明する。

【0046】コマンドワードのRビット(ビット31)は、リセットビットである。マスタプロセッサ60はこのワードをデジタル画像/グラフィックプロセッサに対して発生したり、またはあるデジタル画像/グラフィックプロセッサがこのコマンドワードを自己に発生したりできる。意図している実施例は、いずれのデジタル画像

/グラフィックプロセッサの他のデジタル画像/グラフィックプロセッサをリセットできない。下記のリセットシーケンスの説明全体で、アドレス内の各数字の#はコマンドレジスタCOMM120のビット1~0に記憶されたデジタル画像/グラフィックプロセッサの番号と置換すべきであることに留意されたい。指定されたデジタル画像/グラフィックプロセッサがリセットコマンドワードを受けると、まずホールドラッチをセットし、リセットリクエスト信号を転送コントローラ80へ送る。転送コントローラ80はリセットアクノージング信号をデジタル画像/グラフィックプロセッサへ送る。リセット中のデジタル画像/グラフィックプロセッサは、転送プロセッサ80からリセットアクノージング信号が受信されるまで、別の動作は実行しない。リセットアクノージングの受信時にデジタル画像/グラフィックプロセッサは次のシーケンスのオペレーションを開始する。すなわち既にセットされていなければホールドラッチをセットし、通信レジスタCOMM120のF、P、QおよびSビットを0にクリア(これらビットの使用については後述する)し、アドレスユニット120によりペンディング中のメモリアccessをクリアし、インストラクションキャッシュサービスリクエストをリセットし、インストラクションレジスター実行ステージIRE752に次のインストラクションをロードする。

【0047】

【数3】

```
BR = {u.ncvz} A14 << 1
|| A14 = Hex "0100#7F0"
```

【0048】このインストラクションはプログラムカウンタPC701に対して1ビット左にシフトされたスタックポインタA14の内容に対して変化しないようにされている否定、桁上げ、オーバーフローおよび0ステータスビット、更にRビットセットを無条件にロードし、スタックポインタA14のロードと並行にスタックポインタA14をリセットし、インストラクションレジスタアドレスステージIRAに下記のインストラクションをロードする。

【0049】

【数4】

```
*(PBA + Hex "FC") = PC
```

【0050】このインストラクションはアドレスPBAと16進数FCの合計によって表示されるアドレスにプログラムカウンタPC701の内容を記憶させ、インタラプト疑似インストラクションユニット770をセットして、次にインタラプト疑似インストラクションPS3をロードし、タスクインタラプトを表示するインタラプトフラグレジスタINTFLG115のビット14をセットし、インタラプトフラグレジスタINTFLG115のビット0をクリアしてエミュレータトラップインタ

ラプトETRAPをクリアし、ループ制御レジスタLC TLのビット11、7および3をクリアして、よってすべてのループをディスエーブルする。

【0051】マスタプロセッサ60がアンホールドコマンドワードを送信すると、デジタル画像／グラフィックプロセッサによる実行が開始する。この実行によってデジタル画像／グラフィックプロセッサの作動が開始すると16進数0100#7FCをアドレス指定するよう、プログラムカウンタPC701に記憶されていたアドレスをセーブする。これにより12ビットだけ左にシフトされていったスタックポインタA14の前の内容およびプログラムカウンタPC701の制御ビット（ビット2～0）の現在の値がセーブされ、アドレス16進数010*

*0#7F0をスタックポインタA14にロードし、プログラムカウンタPC701に制御ビット2～0が000となっているタスクインタラプトベクトルをロードし、アドレス16進数0100#7F8に制御ビット2～0を含むインストラクションレジスタアドレスステージIPAの内容を記憶し、アドレス16進数0100#7F4に制御ビット2～0を含むインストラクションレジスタ実行ステージIREの内容を記憶し、タスクインタラプトによって示されたアドレスにおいてプログラムの実行を開始する。表2にリセット後のスタックステートが示されている。

【0052】

【表2】

アドレス	内容
16進数0100#7FC	リセット前の左に1つシフトされた場所からのスタックポインタレジスタA14
16進数0100#7F8	リセット前からのインストラクションレジスタアドレスステージIPA
16進数0100#7F4	リセット前からのインストラクションレジスタ実行ステージIRE

表2

【0053】インストラクションレジスタアドレスステージIPAおよびインストラクションレジスタ実行ステージIREの先のステートは、制御ビット2～0を含む。スタックポインタA14はアドレス16進数0100#7F0を含むことに留意のこと。

【0054】コマンドワードのうちのHビット（ビット30）はホールドビットである。マスタプロセッサ60はこのコマンドワードを任意のデジタル画像／グラフィックプロセッサに発生することもできるし、あるデジタル画像／グラフィックプロセッサはこのコマンドを自らに発生することもできる。意図している実施例では、いずれのデジタル画像／グラフィックプロセッサも他のデジタル画像／グラフィックプロセッサを停止することはできない。指定されたデジタル画像／グラフィックプロセッサがこのコマンドワードを受信すると、デジタル画像／グラフィックプロセッサはホールドラッチをセットし、パイプラインを停止させる。この後のデジタル画像／グラフィックプロセッサは無限にクロスパーメモリが競合しているかのように働く。何もリセットされず、何もインタラプトは起きないか、またはそのように認識される。あるデジタル画像／グラフィックプロセッサはコマンドワードを送ることにより自身を停止させると、ホールドコマンドワードを送るインストラクションの後の2つのインストラクションは、そのインストラクションパイプライン内にあることに留意されたい。ホールドコマンドワードを発生したインストラクションの後の第1インストラクションのアドレスパイプラインステージがインストラクションパイプラインの性質により、すでに

そのアドレスパイプラインステージを実行することになることに留意されたい。このようなホールドステートはマスタプロセッサ60からのアンホールドコマンドワードを受信することによってしか反転できない。

【0055】ホールド条件は変わらないので、デジタル画像／グラフィックプロセッサ内のパワー消費量を減少させる。デジタル画像／グラフィックプロセッサがこのモードになっている間、クロックを停止することにより更にパワーを節約できる。

【0056】コマンドワードのUビット（ビット29）は、アンホールドビットである。このコマンドワードはマスタプロセッサ60から一つ以上のデジタル画像／グラフィックプロセッサ71、72、73、74によってしか発生できない。アンホールドコマンドワードは宛て先デジタル画像／グラフィックプロセッサのホールドラッチをクリアし、デジタル画像／グラフィックプロセッサは次に、何も起こらなかったがごとくホールドの後にコードの実行を再開する。これはハードウェアまたはコードリセットの後でデジタル画像／グラフィックプロセッサをスタートさせる好ましい方法である。アンホールドコマンドワードの実行時に宛て先デジタル画像／グラフィックプロセッサは、そのタスクインタラプトベクトルによって示されたアドレスでもコードの実行を始める。Uビットの優先度は単一コマンドワードのHビットよりも高い。したがってHビットとUビットとの双方を備えた単一コマンドワードを受信する結果、アンホールドコマンドが実行される。マスタプロセッサ60からのアンホールドコマンドワードとデジタル画像／グラフィ

ックプロセッサ自体により送信されるホールドコマンドワードを同時に受信することにより、マスタプロセッサ60のアンホールドコマンドワードに優先権が与えられる。これによりRビットはUビットより優先度が高くなる。従って、RビットとUビットセットの双方を有するマスタプロセッサ60からの単一コマンドワードが受信される結果、デジタル画像／グラフィックプロセッサはホールド状態にリセットされる。

【0057】コマンドワードのIビット（ビット28）は、インストラクションキャッシュフラッシュビットである。マスタプロセッサ60がこのコマンドワードをいずれかのデジタル画像／グラフィックプロセッサに送ってもよいし、一つのデジタル画像／グラフィックプロセッサがかかるコマンドワードを自身に発生してもよい。意図している実施例では、いずれのデジタル画像／グラフィックプロセッサも他のデジタル画像／グラフィックプロセッサによるインストラクションキャッシュフラッシュを命令できない。このコマンドワードを受信する指定されたデジタル画像／グラフィックプロセッサはそのインストラクションキャッシュをフラッシュする。インストラクションキャッシュフラッシュは、キャッシュタグの値フィールドをキャッシュタグレジスタ自身の番号にセットさせ、存在するビットのすべてをクリアし、L、R、Uビットをタグレジスタ自身の番号にセットする。

【0058】コマンドワードのDビット（ビット27）は、データキャッシュフィールドを表示する。デジタル画像／グラフィックプロセッサ71、72、73、74は、データキャッシュを使用しないので、このコマンドワードはデジタル画像／グラフィックプロセッサに適用されず、これらに無視される。マスタプロセッサ60はこのコマンドワードを自らに送り、そのデータキャッシュメモリ13および14をフラッシュさせることができる。

【0059】コマンドワードのKビット（ビット14）は、タスクインタラプトを表示している。マスタプロセッサ60はこのコマンドワードをいずれのデジタル画像／グラフィックプロセッサ71、72、73または74にも送ることができるが、いずれのデジタル画像／グラフィックプロセッサも他のデジタル画像／グラフィックプロセッサまたはマスタプロセッサ60にこのコマンドワードを送ることはできない。

【0060】コマンドワードのGビット（ビット13）は、メッセージインタラプトを表示する。いずれのデジタル画像／グラフィックプロセッサもこのメッセージインタラプトを他のデジタル画像／グラフィックプロセッサまたはマスタプロセッサ60に送ることができる。かかるコマンドワードで指定されたデジタル画像／グラフィックプロセッサは、そのメッセージインタラプトフラグをセットし、メッセージインタラプトがインタラプト

イネーブルレジスタINTEN110のビット20によりイネーブルされると、メッセージインタラプトを取り込む。好ましい実施例では、このコマンドワードは転送コントローラ80に送られることはない。

【0061】デジタル画像／グラフィックプロセッサが自身にコマンドワードを発生し、Hビットにより自らを停止させたり、またはIビットによってインストラクションキャッシュをフラッシュさせると、このコマンドを実行するには、このコマンドワードは対応するデジタル画像／グラフィックプロセッサの指定ビットセットを有していなければならない。これは一貫性を持たせ、かつコマンドワード機能を将来的に拡張できるようにするためである。

【0062】図5は通信レジスタCOMMのフィールド定義を略図で示す。F、S、QおよびPビット（ビット31～28）は、デジタル画像／グラフィックプロセッサ71、72、73または74および転送コントローラ80からのパケット転送の通信に使用される。FおよびSビットは通常の読み出し／書き込みビットであり、PビットはSビットが0であるか、または同時に0にクリアされる場合に限り書き込みが可能である。9ビットは、読み出し専用であり、パケット転送は転送コントローラ80によるデータ移動のためのデジタル画像／グラフィックプロセッサ71、72、73または74によるリクエストである。これらデータ移動ではマイクロプロセッサ集積回路100の内部のメモリ11～14および21～40だけを必要とする場合もあれば、内部メモリと外部メモリの双方を必要とする場合もある。パケット転送はリンクされたリスト構造として記憶され、各デジタル画像／グラフィックプロセッサに対しては、一度に一つのパケット転送しかアクティブにできない。リクエスト中のデジタル画像／グラフィックプロセッサ7172、73または74に対応するパラメータメモリ25、30、35または40内の専用アドレスにおけるリンクされたリストポインタは、アクティブなリンクされたリストの開始点をポイントする。リンクされたリストにおける各エントリーは、次のリストエントリーに対するポインタを含む。

【0063】パケット転送の初期化を行うには次のステップが必要である。まずデジタル画像／グラフィックプロセッサは、対応するパラメータメモリに所望するパケット転送パラメータをセットする。次にデジタル画像／グラフィックプロセッサは対応するパラメータメモリに所定のアドレスの16進数0100#0FCにリンクされたリストの第1リンクのアドレスを記憶し、ここで#はデジタル画像／グラフィックプロセッサの番号と置換される。Pビット（ビット28）の1へのセットにより、パケット転送の転送コントローラ80に警告が与えられる。デジタル画像／グラフィックプロセッサはFビット（ビット31）を1にセットすることにより高い優

先度をリクエストしたり、またはFビットをクリアすることにより低い優先度をリクエストできる。

【0064】転送コントローラ80はPビットがセットされたことを認識し、Fビットのステートに基づきパケット転送への優先度を割り当てる。転送コントローラ80はPビットをクリアし、Qビットをセットすることによりパケット転送が待ち行列内にあることを表示する。次に転送コントローラ80は対応するパラメータメモリ内の所定のアドレス、すなわち16進数の0100#0FCにアクセスし、リンクされたリストに基づくパケット転送のサービスを行う。パケット転送が完了すると、転送コントローラ80はQビットを0にクリアし、待ち行列がもはやアクティブでないことを表示する。デジタル画像／グラフィックプロセッサは、パケット転送が完了したかどうかを表示するため、このビットを周期的に読み出すことができる。これとは別に、パケット転送自体がパケット転送終了時にリクエスト中のデジタル画像／グラフィックプロセッサをインタラプトするように転送コントローラ80に命令することもできる。この場合、転送コントローラ80はビット19すなわちパケット転送エンドインタラプトビットPTENDをインタラプトフラグレジスタINTFLG115にセットすることにより、インタラプトをデジタル画像／グラフィックプロセッサへ送る。パケット転送サービス中に転送コントローラ80がエラーを生じると、ビット18、すなわちパケット転送エラーインタラプトビットPTERRORをインタラプトフラグレジスタINTFLG115にセットすることにより、デジタル画像／グラフィックプロセッサにインタラプト信号を送る。デジタル画像／グラフィックプロセッサは表1および適当なインタラプトサービスルーチンに記載された位置に記憶されている適当なインタラプトベクトルを有する。

【0065】デジタル画像／グラフィックプロセッサは、転送コントローラ80が先のリクエストにサービスしている間、別のパケットをリクエストできる。この場合、デジタル画像／グラフィックプロセッサはQビットが1である間にPビットを1にセットする。これが行われると、転送コントローラ80はインタラプトフラグレジスタINTFLG115のビット17をセットすることにより、デジタル画像／グラフィックプロセッサにパケット転送ビジーインタラプトPTBを送る。転送コントローラ80はPビットを0にクリアする。リクエスト中のデジタル画像／グラフィックプロセッサのインタラプトサービスルーチンは、第1パケット転送が待ち行列内にある間、第2パケット転送を一時中止し、パケット転送をキャンセルしたり、他のある訂正措置をとることができる。このような特徴により、デジタル画像／グラフィックプロセッサは通信レジスタCOMM120のQビットを最初にチェックすることなく、パケット転送を送ることができる。

【0066】デジタル画像／グラフィックプロセッサは、Sビットを1にセットすることによりパケット転送のサービスを一時中止にできる。転送コントローラ80は、Sビットが1であることを検出する。パケット転送が待ち行列内にある間このことが起きると、転送コントローラ80はQビットをPビットにコピーし、Qビットをクリアする。これによりPビットは一般に1にセットされる。リクエスト中のデジタル画像／グラフィックプロセッサ内のソフトウェアは、SおよびPビットのステータスを変えることができる。転送コントローラ80は一時中断されたパケット転送のリンクされたリスト内の位置をメモリに保持する。Sビットが0であり、Pビットが同時に1であると、転送コントローラ80が判断すると、一時中止されていたパケット転送は再開される。

【0067】通信レジスタCOMM120の同期ビットフィールド（ビット15～8）は、同期したマルチインストラクション、マルチデータモードで使用される。このフィールドは、同期したマルチインストラクション、マルチデータモードを可能にするロックインストラクションLCKおよびこのモードをディスエーブルするアンロックインストラクションUNLCKによって境界が定められたインストラクションのために作動する。ビット11～8は、インストラクションのフェッチをデジタル画像／グラフィックプロセッサ74、73、72および71とそれぞれ同期化すべきかどうかを表示している。これらビットのいずれかにおける1は、対応するデジタル画像／グラフィックプロセッサが先のインストラクションの実行を完了したことを表示するまでのデジタル画像／グラフィックプロセッサがインストラクションフェッチを遅延することを示している。このデジタル画像／グラフィックプロセッサを同期すべき他のデジタル画像／グラフィックプロセッサは、同様にして通信レジスタCOMM120内の対応するビットをセットすることにある。デジタル画像／グラフィックプロセッサが同期したマルチインストラクション、マルチデータモードにある場合、それ自身に対応する同期ビットをセットする必要はないが、セットしても害はない。ビット15～12は、8つのデジタル画像／グラフィックプロセッサまで拡張する場合に保留したものである。

【0068】通信レジスタCOMM120のDIGP#フィールド（ビット2～0）は、マルチプロセッサ集積回路100上の各々の特定のデジタル画像／グラフィックプロセッサに対してユニークとなっている。これらビットは読み出し専用であり、これらビットに書き込もうとする試みは不可能である。これは同地つでないデジタル画像／グラフィックプロセッサ71、72、73および74の一部にすぎないからであり、ビット1～0は表3に示すような特定のデジタル画像／グラフィックプロセッサを識別する2ビットのコードに配線が発生できる。

【0069】

【表3】

COMM フィールド		パラレル プロセッサ
1	0	
0	0	DIGP0 (71)
0	1	DIGP1 (72)
1	0	DIGP2 (73)
1	1	DIGP3 (74)

【0070】ビット2は8個のデジタル画像／グラフィックプロセッサを有するマルチプロセッサ集積回路100で使用するために保留されていることに留意された。現在好ましい実施例は、このビットは、4つのデジタル画像／グラフィックプロセッサ71、72、73および74のすべてに対し0となるように、配線で発生される。

【0071】通信レジスタCOMM120のこの部分は特定のデジタル画像／グラフィックプロセッサを識別するように働き、デジタル画像／グラフィックプロセッサの識別番号は通信レジスタCOMM120と7（16進数0000007）とAND論理演算することによって抽出できる。例えば、インストラクションD0=COMM&7がこれを行う。このインストラクションは通信レジスタCOMM120のビット2～10内のデータだけをリターンさせる。このインストラクションは8個のデジタル画像／グラフィックプロセッサを有する実施例に対して適しており、各デジタル画像／グラフィックプロセッサに対応するデータメモリおよびパラメータメモリのアドレスは、そのデータ画像／グラフィックプロセッサの識別に依存するので、識別番号によりソフトウェアがこれらに対応するメモリのアドレスを計算できる。このような識別番号を使用することにより、プログラムを実行する特定のデジタル画像／グラフィックプロセッサと独立したソフトウェアを書き込むことが可能となる。デジタル画像／グラフィックプロセッサと独立したプログラムが、対応するパラメータメモリのベースアドレスと、データメモリのベースアドレスに対しレジスタPBAおよびDBAを使用することもできる。

【0072】マルチプロセッサ集積回路100は小エンディアンまたは大エンディアンフォーマットで作動できる。特に表記しなければ、図および表は大エンディアンフォーマットでのオペレーションを示す。内部レジスタおよび外部データバスのためのビットナンバリングは、常に右側にビット0を有する小エンディアンの規定に従う。64ビットのワード内のバイトは、小エンディアンモードでは右側からアドレス指定され、大エンディアンモードでは左側からアドレス指定される。常に32ビットオペレーションを用いて内部レジスタにアクセスすることにより混乱を受けることができる。他の実施例でバス幅を広くしたり狭くしたりすることもできる。16ビットの整数倍であるバス幅が最も有効であると信じられる。

【0073】転送コントローラ80は、マルチプロセッサ集積回路100のうちのマスタプロセッサ60およびデジタル画像／グラフィックプロセッサ71、72、73、74と外部メモリ、例えばビデオRAM5および6並びにメモリ9との間のインターフェースとなっている。転送コントローラ80は、いくつかの自律的メモリオペレーションのみならずプロセッサによってリクエストされるメモリオペレーションを実行する。これらについては後述する。転送コントローラ80はすべてのメモリ11、12、13、14、15、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39および40にアクセスをするクロスバー50に、64ビットのバス接続をしている。従って転送コントローラ80はインストラクションキャッシュ、データメモリおよびパラメータメモリのすべてにアクセスできる。第2の64ビットのバスは、画像システムバスに接続し、オフチップアクセスを行う。

【0074】転送コントローラ80はそのホールド／ホールドアクノージング機構を使用することにより、画像システムを介する外部メモリへのアクセスを外部ホストが行うことを可能にしている。転送コントローラ80はキャッシュミスおよびパケット転送リクエストを行い、外部DRAM/VRAMのリフレッシュを行い、フレームコントローラ90が必要とするシリアルレジスタ転送サイクルを実行し、VRAMをベースとするディスプレイ／キャプチャバッファを更新する。メモリ22、23、24、27、28、29、32、33、34、37、38および39を用いる転送コントローラ80のメモリオペレーションは、通常、デジタル画像／グラフィックプロセッサ71、72、73、74またはマスタプロセッサ60からのパケット転送リクエストに回答して行われる。パケット転送は、オンチップメモリとオフチップメモリとの間でデータを転送する極めてフレキシブルな方法を提供するものである。転送コントローラ80は直接外部アクセス（DEA）サイクルを用いることにより、プロセッサに直接データを送ることもできる。直接外部アクセスサイクルはデジタル画像／グラフィックプロセッサ71、72、73、74がオフチップメモリにアクセスできるようにすると共に、外部メモリにアクセスする際にマスタプロセッサ60がそのデータキャッシュメモリをバイパスできるようにする。転送コントローラ80は種々のリクエストの優先権を定め、必要であればパケット転送リクエスト間で外部メモリインターフェースをタイムシェアする。マスタプロセッサ60、デジタル画像／グラフィックプロセッサ71、72、73、74、フレームコントローラ90およびホスト処理システム1からのリクエストのすべては固定された優先度（順位）決定方法によりサービスされる。同じ優先度の多数のリクエストがペンディングであると、転送コン

トローラ80はラウンドロビン方法に基づいてこれらにサービスを行う。

【0075】転送コントローラ80はプロセッサからの多くの異なるタイプのリクエストを処理しなければならない。最適なシステムの性能を保つには、これらリクエストは緊急度および重要性によって優先度が定められる。転送コントローラ80はこれらの異なる優先度に基づいて作動するので、クロスパー50に対する自己の優先度はサイクルごとに変わり得る。

【0076】図6は、画像システムバス上での転送コントローラ80のオペレーションの優先度を示す。これは階層200として示されている。同じ優先度の多数のリクエストが受信されると、転送コントローラ80はこれらをラウンドロビン法で処理する。これについては図6に示されている。いずれのプロセッサもパケット転送に対して一つのアクティブな優先度しか有することができない。マスタプロセッサ60が緊急優先度、高い優先度および低い優先度のパケット転送リクエストを送ることができるが、デジタル画像/グラフィックプロセッサ71、72、7374は、高低優先度のパケット転送に限定されている。

【0077】最高優先度201はホスト処理システム1による外部バスリクエスト(HRRQ)のサービスである。転送コントローラ80はホストリクエストライン上の信号に応答し、画像システムバスの制御をホスト処理システム1の下に置くことができる。

【0078】次に低い優先度202は、フレームコントローラ90からのメモリリクエストのサービスである。次に低い優先度203は、緊急なダイナミックランダムアクセスメモリ(DRAM)のリフレッシュリクエストのサービスである。更に下に述べるように、DRAMリフレッシュリクエストの所定のバックログが形成された際に、これら緊急DRAMリフレッシュリクエストが生じる。

【0079】次の優先度204は、マスタプロセッサ60のインストラクションキャッシュメモリ11および12、データキャッシュメモリ12および13のサービス並びにインタラプトがディスエーブルされたときのマスタプロセッサ60の緊急パケットリクエストである。各プロセッサに含まれる関連するキャッシュロジックにより、キャッシュに関連する転送コントローラ80のすべてのオペレーションが自動的にリクエストされる。リクエストされたデータは外部メモリ位置からリクエスト中のプロセッサのデータキャッシュメモリ内の適当なサブブロックへ移動される。転送コントローラ80はマスタプロセッサ60が必要とする際に、マスタプロセッサ60のためにデータキャッシュメモリ13および14のデータサブブロックもセーブする。緊急パケット転送リクエストはマスタプロセッサ60によってリクエストで
きただけであり、パケット転送リクエスト内の特定のビ

ットのセッティングを含む。これについては後述する。マスタプロセッサ60がインタラプトをイネーブルする場合、これらデータ転送は優先度が低いことに留意されたい。

【0080】次に低い優先度は、デジタル画像/グラフィックプロセッサ71、72、7374またはマスタプロセッサ60のインストラクションキャッシュメモリ11および12、データキャッシュメモリ12および13のインストラクションキャッシュリクエストまたはダイレクト外部アクセス(DEA)リクエストのサービス、並びにインタラプトがイネーブルされている時のマスタプロセッサ60の緊急パケット転送リクエストを行うことである。ここで、同じレベルの同時のアクセスリクエストは、リクエストしているプロセッサに応じてラウンドロビン法により処理されることに留意されたい。同じプロセッサに対するキャッシュサービスリクエストおよびダイレクト外部アクセスリクエストは、次のプロセッサにトークンを送る前にサービスされる。

【0081】次の優先レベル206は、高い優先度のパケット転送である。図6に示すように、種々のプロセッサから生じたパケット転送は、ラウンドロビン法で処理される。マスタプロセッサ60またはデジタル画像/グラフィックプロセッサ71、72、73または74のうちの一つがリクエストされたデータを待っている場合、通常、高い優先度のパケット転送を用いる。転送コントローラ80は次の優先レベル207を有する低い優先度のパケット転送に対して同様なラウンドロビン法を用いる。マスタプロセッサ60またはデジタル画像/グラフィックプロセッサ71、72、73または74のうちの一つが、リクエストされたデータを待っていない場合、このプロセッサは低い優先度のパケット転送を通常用いる。後述するように、リクエスト中のプロセッサはパケット転送を緊急にするか、高い優先度にするか、または低い優先度にするかを表示する。

【0082】各リクエストの優先原理について、下に説明する。外部デバイス、例えばホスト処理システム1は、欲する場合、迅速なアクセスを行うことができない。外部デバイスは後述するように、REQ[1:0]をモニタすることにより、必要であれば高い優先度のリクエストに対し転送コントローラ80に画像システムバスを戻すことができる。フレームコントローラ90のリクエストはビデオディスプレイまたはビデオキャプチャを悪化することなく、タイムクリティカルなVRAM転送サイクルが生じるように、第2の優先度を受け入れる。ホストリクエストサイクルおよびフレームコントローラ90のリクエストは、間欠的にしか発生しないので、高い優先度を必要とする緊急DRAMリフレッシュはその下の優先度が与えられる。次の優先度はマスタプロセッサ60のキャッシュサービス、ダイレクト外部アクセスおよびマスタプロセッサ60のインタラプ

トがディスエーブルされている際に生じる緊急優先度の
 パケット転送サイクルである。これにより、通常インタ
 ラプトをディスエーブルするマスタプロセッサ60のイン
 タラプトサービスルーチンは、システム性能を最大に
 するようにできるだけ迅速に実行することができる。デ
 ジタル画像／グラフィックプロセッサ71、72、7
 3、74のキャッシュサービスおよびダイレクト外部ア
 クセスリクエストは次の優先度である。リクエストのサ
 ービスがなされるまで、プロセッサはアイドル状態であ
 るので、これらを迅速にサービスすることが重要であ
 る。マスタプロセッサ60のキャッシュサービス、ダイ
 レクト外部アクセスリクエストおよび緊急パケット転送
 も、インタラプトがイネーブルされている場合のこの優
 先度にある。その理由は、マスタプロセッサ60はシス
 テム内の他の場所からのインタラプトリクエストにサー
 ビスしていないからである。高い優先度のパケット転送
 はリクエスト中のプロセッサが転送を終了するのにデー
 タを待っていること、または転送コントローラ80が外
 部バスバンド幅を最適にするよう、クロスバーアクセス
 のためにデジタル画像／グラフィックプロセッサ71、
 72、73および74よりも高い優先度とすることが必要
 であることを意味している。低い優先度のパケット転
 送は、プロセッサがデータを待っていないことを意味し
 ているので、これらには極めて低い優先度が与えられ
 る。意図する実施例では、トリックルリフレッシュサイ
 クルに最下位の優先度が与えられる。これらサイクル
 は、外部バスがアイドル状態であり、リフレッシュバッ
 クログが0でない場合に実行されるだけである。これ
 は、バックログを小さくし、後の時間に高い優先度の緊
 急リフレッシュがリクエストされる可能性を少なくする
 のに役立っている。

【0083】転送コントローラ80が同じ優先度の異な
 るプロセッサから多数のリクエストを受けるときはいつ
 も、これらの中でラウンドロビン法を行う。このラウン
 ドロビン法は、固定された周期的な優先方法である。す
 なわちこのことは、ラウンドロビンからいずれのプロセ
 ッサも除くことができず、サイクル内のプロセッサの順
 序を変えることができないことを意味している。特定の
 プロセッサのリクエストが完了すると、常にチェーン内
 でペンディング中のリクエストを有する次のプロセッサ
 にラウンドロビントークンが渡される。他のプロセッサ
 からの等しい優先度のリクエストをサービスする必要が
 あるときは、これによって一つのプロセッサが転送コン
 トローラ80を独占することが防止される。

【0084】転送コントローラ80のサービスを行うリ
 クエストレベルに応じ、転送コントローラ80のクロス
 バーの優先度はダイナミックに変わる。これについて
 は、図7において階層210として示されている。転送
 コントローラ80には次の優先度が割り当てられる。緊
 急優先度のパケット転送リクエスト、キャッシュサービ

スリクエスト、ダイレクト外部アクセスリクエストのサ
 ービスを行っているとき、またはそのパイプラインをフ
 ラッシングしているときに、転送コントローラ80はマ
 スタプロセッサ60の優先度212の上の優先度で作動
 する。転送コントローラ80が、緊急DRAMリフレッ
 シュリクエスト、フレームコントローラ90からのリク
 エスト、ホストインターフェースリクエストまたはソフ
 トリセットを受けるときはいつも、そのパイプラインを
 フラッシュする。これらは完了を待っている待機中の外
 部サイクルで開始することはできない。パイプラインの
 フラッシングは、ときどき生じるか、または短時間の間
 マスタプロセッサ60をロックアウトするだけである。

【0085】転送コントローラ80にはデジタル画像／
 グラフィックプロセッサ71、72、73、74の優先
 度214よりも高く、かつ高優先度のパケット転送用マ
 スタプロセッサ60よりも低い優先度213が与えられ
 る。これにより、マスタプロセッサ60をロックアウト
 することなく転送コントローラ80に最大の可能な優先
 度が与えられる。マスタプロセッサ60は一般にコント
 ローラとして使用されるので、長期間これをロックアウ
 トすることは、システムに好ましくない影響が及ぶ。

【0086】転送コントローラ80が低い優先度のパケ
 ット転送を行う場合、このコントローラの優先度214
 はデジタル画像／グラフィックプロセッサ71、72、
 73、74のラウンドロビンの優先度214よりも低い。
 これによってパケット転送の優先度が低い際に転送
 コントローラ80がデジタル画像／グラフィックプロセ
 ッサ71、72、73、74からのクロスバンド幅をス
 チールすることが防止される。デジタル画像／グラフィ
 ックプロセッサ71、72、73、74のローカルメモ
 リアクセスは異なるメモリに向けられ、干渉できないこ
 とに留意すべきである。従ってこれらローカルメモリア
 クセスはラウンドロビン法で同じステージを有する。

【0087】転送コントローラ80によってより高い優
 先度のリクエストが受け入れられる場合、このコントロ
 ーラは新しいリクエストのクロスバーの優先度で現在の
 オペレーションを完了したり、一時中断したりする。こ
 れによりシステム内でのブロックが発生しないよう保証
 される。従って例えば高い優先度のパケット転送リクエ
 ストが受け入れられる場合、高い優先度で低い優先度の
 パケット転送の一時中断が生じる。

【0088】転送コントローラ80はデジタル画像／グ
 ラフィックプロセッサ71、72、73、74のインス
 トラクションキャッシュミス、マスタプロセッサ60の
 インストラクションおよびデータキャッシュミスに自動
 的にサービスを行う。多数のキャッシュサービスリクエ
 ストが受信されると、転送コントローラ80は図6に示
 すように、ラウンドロビン法に基づき、これらの優先度
 を定める。キャッシュミスのサービスが完了すると、転
 送コントローラ80はリクエスト中のプロセッサにシグ

10

20

30

40

50

ナルを送る。マスタプロセッサ60はラウンドロビンでの順番内にサービスされるインストラクションおよびデータキャッシュの双方を有することができる。

【0089】デジタル画像／グラフィックプロセッサ71、72、73、74のインストラクションキャッシュは、各々4つの128バイト（16インストラクション）のサブブロックを含む4つのブロックから成る1方向の組の関連（完全関連）キャッシュである。他の実施例では、他方向の組の関連キャッシュを用いることもできる。デジタル画像／グラフィックプロセッサがキャッシュミスを経験すると、このプログラムフロー制御ユニットは転送コントローラ80に信号を送ってキャッシュミスをサービスする。デジタル画像／グラフィックプロセッサ71、72、73または74は、インストラクションを入れるキャッシュブロックを決定し、この情報のみならずアドレスも転送コントローラ80へ送る。転送コントローラ80は、外部メモリからの完全サブブロック（128バイト）をフェッチし、これをリクエスト中のデジタル画像／グラフィックプロセッサ71、72、73、74の適当なキャッシュサブブロックへ入れる。次に転送コントローラ80はリクエストのサービスが完了し、プロセッサがそのプログラムの実行を続けることができる旨を、デジタル画像／グラフィックプロセッサ71、72、73または74へ知らせる。

【0090】マスタプロセッサ60は、各々4ブロックから成る4方向の関連する組である。各ブロックは4つの64バイトのサブブロックを含む。マスタプロセッサ60はインストラクションキャッシュまたはデータキャッシュ、または双方に対するサービスをリクエストできる。マスタプロセッサ60のインストラクションキャッシュサービスリクエストは、転送コントローラ80によりフェッチされるサブブロックの大きさがわずか64バイトであることを除けば、デジタル画像／グラフィックプロセッサ71、72、73、74のキャッシュリクエストと同様に取り扱われる。

【0091】マスタプロセッサ60のデータキャッシュメモリ13および14は、転送コントローラ80がその内容を外部メモリに書き戻すようにリクエストできる点でインストラクションキャッシュ11および12と異なっている。転送コントローラ80はマスタプロセッサ60のデータキャッシュミスに対し、マスタプロセッサ60のインストラクションキャッシュを用いる場合と同じように、64バイトのサブブロックをフェッチする。しかしながら一致するタグアドレスが発見できず、すべてのブロックを使用したことによりマスタプロセッサ60がブロックミスを起こす場合、このプロセッサはまず最初に転送コントローラ80にブロックを置換する前に、最後に使用したブロック内にダーティサブブロックを書き戻すことをリクエストする。ダーティサブブロックの書き戻しは、ラウンドロビン優先度内の単一マスタプロ

セッサ60の順番内で行うことができる。転送コントローラ80は特殊キャッシュインストラクションに 응답してダーティサブブロックを書き戻すようにもリクエストできる。

【0092】転送コントローラ80はマスタプロセッサ60およびデジタル画像／グラフィックプロセッサ71、72、73、74からのすべてのダイレクト外部アクセス（DEA）リクエストを処理する責任を負っている。ダイレクト外部アクセスサイクルはデジタル画像／グラフィックプロセッサ71、72、73、74が外部メモリ内のデータに直接アクセスできるようにすると共に、マスタプロセッサ60がそのデータキャッシュをバイパスできるようにする。ダイレクト外部アクセスには高い優先度が与えられているので、これらアクセスは長さが1バイト、半ワード（16ビット）、1ワード（32ビット）、またはダブルワード（64ビット）となり得る単一アクセスに限られる。これにより、一つのプロセッサが多数のダイレクト外部アクセスサイクルで外部バスを独占することが防止され、これによりダイレクト外部アクセスリクエストおよび他のプロセッサのキャッシュミスのサービスが行われるのが防止される。単一のオフチップメモリ位置への高速アクセス、例えばプログラム可変またはオフチップレジスタが必要なときに、ダイレクト外部アクセスサイクルが使用される。

【0093】デジタル画像／グラフィックプロセッサ71、72、73、74は、それらのパラメータメモリ25、30、35および40、並びにそれらのデータメモリ22、23、24、26、28、29、32、33、34、37、38および39に通常アクセスする。16進数02000000以上のアドレスへのアクセスにより、ダイレクト外部アクセスリクエストは自動的に転送コントローラ80へ送られる。このリクエストはキャッシュダイレクト外部アクセスラウンドロビンにおけるリクエスト中のデジタル画像／グラフィックプロセッサ71、72、73、74の順番に達したときにサービスを受ける。キャッシュミスリクエストおよびダイレクト外部アクセスリクエストの双方は、継続中の場合に1回でサービスを行うことができる。クロスバーを介してアクセスできないオンチップメモリ領域、例えばマスタプロセッサ60のパラメータメモリ16へのデジタル画像／グラフィックプロセッサ71、72、73、74のアクセスは、転送コントローラ80へのダイレクト外部アクセスリクエストに変換される。しかしながらこのダイレクト外部アクセスはフォールトとなる。フォールトを生じさせるようなデジタル画像／グラフィックプロセッサ71、72、73、74のダイレクト外部アクセスサイクルは、フォールトしたデジタル画像／グラフィックプロセッサ71、72、73、74のキャッシュサイクルと同じように取り扱われる。

【0094】マスタプロセッサ60はデジタル画像／グ

グラフィックプロセッサ71、72、73、74と若干異なる態様でダイレクト外部アクセスサイクルを用いる。マスタプロセッサ60は、データキャッシュメモリ13および14により通常外部メモリにアクセスする。マスタプロセッサ60はダイレクト外部アクセスサイクルを利用し、データキャッシュメモリ13および14およびアクセスメモリを直接バイパスする。特殊メモリロードまたは記憶インストラクションを用いることにより、ダイレクト外部アクセスサイクルが明瞭に指定される。マスタプロセッサ60にアクセスできないオンチップアドレス、例えばデジタル画像／グラフィックプロセッサ71、72、73、74のインストラクションキャッシュメモリ21、26、31および36に対するかかる特別ロードまたは記憶インストラクションを、マスタプロセッサ60が実行しようとする場合、このオペレーションはダイレクト外部アクセスリクエストに変換され、次に変換コントローラ80によってフォールトされる。アクセス可能なオンチップメモリ領域、例えばオンチップレジスタまたはデジタル画像／グラフィックプロセッサ71、72、73、74のデータメモリ22、23、24、27、28、29、32、33、34、37、38、39への特別メモリロードまたは記憶オペレーションは、通常のロードまたは記憶オペレーションに変換され、転送コントローラ80を介してダイレクト外部アクセスリクエストは行われない。

【0095】図8は、転送コントローラ80の高度なブロック図を示す。下記に、各主要ブロックを簡単に説明する。転送コントローラ80は内部メモリインターフェース301と、外部メモリインターフェース302と、リクエスト待ち行列化および優先度決定回路303と、キャッシュ、VRAMおよびリフレッシュコントローラ310と、パケット転送用先入れ先出し(FIFO)バ

ッファ311と、キャッシュバッファ312と、ソースレジスタ321およびソース制御ロジック322から成るソースマシン320と、ソースマルチプレクサおよびアライメントロジック330と、宛て先レジスタ341および宛て先制御ロジック342から成る宛て先マシン340と、宛て先マルチプレクサおよびアライメントロジック302とから成る。

【0096】転送コントローラ80はマルチプロセッサ集積回路100による外部メモリへのすべてのアクセスに対して責任を負っている。外部メモリインターフェースは多数の異なるタイプや大きさのメモリデバイスおよび周辺機器に合わせるように設計されている。アクセスされるメモリのタイプは、ダイナミックに決定されるので、各メモリサイクルのタイミングはアクセスされるデバイスに対して最適にすることができる。

【0097】次は、外部メモリアインターフェース302を介して外部データアクセスのために使用されるマルチプロセッサ集積回路100の信号のリストである。

【0098】アドレスバスA[31:0]。これはマルチプロセッサ集積回路100からの32ビットのバイトアドレスを外部メモリに送るためのものである。このアドレスはDRAMアクセスに対して多重化できる。

【0099】アクセスシフト選択AS[2:0]。これらの入力値は転送プロセッサ80による別アドレスへ与えられるシフト量を決定する。表4aおよび4bに示すように、0を含む8つのシフト量がサポートされている。表4aはアドレスバスビットA[31:16]上の出力を示し、表4bはアドレスバスビットA[15:0]上の出力を示す。

【0100】

【表4】

AS[2:0]	A[31:16]															
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0 0 0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0 0 1	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8
0 1 0	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7
0 1 1	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6
1 0 0	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5
1 0 1	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4
1 1 0	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3
1 1 1	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2

表 4a

AS[2:0]	A[15:0]															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0 0 0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0 0 1	7	6	5	4	3	2	1	0	X	X	X	X	X	2	1	0
0 1 0	6	5	4	3	2	1	0	X	X	X	X	X	X	2	1	0
0 1 1	5	4	3	2	1	0	X	X	X	X	X	X	X	2	1	0
1 0 0	4	3	2	1	0	X	X	X	X	X	X	X	X	2	1	0
1 0 1	3	2	1	0	X	X	X	X	X	X	X	X	X	2	1	0
1 1 0	2	1	0	X	X	X	X	X	X	X	X	X	X	2	1	0
1 1 1	1	0	X	X	X	X	X	X	X	X	X	X	X	2	1	0

表 4b

【0101】行アドレス時間の間、アドレスバスA [3 20*
1:0] は、通常のアドレス値を出力する。列アドレス
時間の間、アドレスバスA [31:0] は表4aおよび
4bに示すようにアドレスシフト選択AS [2:0] に
従ってシフトされたアドレス値を出力する。Xと表示さ
れたアドレスライン出力は、ドライブされず、高インピ
ーダンス状態にセットされたものであることに留意され
たい。

【0102】バスサイズ選択BS [1:0]。これは6
4ビット幅よりも狭いデータバスに対するダイナミック
なバスの大きさの決定を可能とする。下記の表5に、こ
れら入力信号のコード化を示す。

【0103】

【表5】

BS[1:0]		ビット サイズ
0	0	8 ビット
0	1	16 ビット
1	0	32 ビット
1	1	64 ビット

【0104】出力ブロックCLKOUT。このクロック
出力は外部ロジックがマルチプロセッサ集積回路100
の作動に同期できるようにするものである。別タイミン
グ選択CT [1:0]。これらの入力信号は転送コント
ローラ80によって与えられる現在のメモリサイクルの
どのタイミングであるかを決定する。下記の表6にこれ
ら入力信号のコード化を示す。

【0105】

【表6】

CT [1:0]		メモリアイミング
0	0	パイプライン状の1サイクル/列
0	1	非パイプライン状の1サイクル/列
1	0	非パイプライン状の2サイクル/列
1	1	非パイプライン状の3サイクル/列

表 6

【0106】列アドレスストローブCAS' [7:
0]。これら出力は、DRAM/VRAMの反転CAS
をドライブする。個々のバイトアクセスを可能にするよ
うに、8個のストローブが達成される。エンディアンモ
ードにかかわらず、反転CAS [0] は、D [7:0]
でのデータ転送に対応し、CAS' [1] は、D [1
5:9] 上の転送に対応する。トランスペアレンシーで
はこれら信号はバイト書き込みストローブとして使用さ
れる。データバスD [63:0]。この入出力バスは、
マルチプロセッサ集積回路100の内外にメモリサイク
ル当たり64ビットまでのアクセスを可能とする。

【0107】データバッファ出力イネーブルDBE
N'。この出力はデータトランシーバをオンにするのに
用いることができる。データ方向インディケータDDI
N'。この出力は、データトランシーバに対する方向イン
ディケータとなる。特別機能ピンDSF1およびDS
F2。これら出力は特別VRAM機能を選択するのに使
用される。フォールト反転FAULT。この入力メモリ
フォールトが生じたことを、マルチプロセッサ集積回
路100に知らせる。

【0108】ページサイズ選択PS [2:0]。これら
入力は現在アクセスされているメモリのページ選択をマ

ルチプロセッサ集積回路100に示す。これら入力の使用については、後に更に詳細に説明する。行アドレストローブ反転RAS。これら出力はDRAM/VRAMの反転RAS入力をドライブする。

【0109】READY。この入力は完了すべきメモリサイクルに対して外部デバイスがレディー状態となっていることを表示する。これはメモリサイクル内に待機ステートを挿入するのに、転送コントローラ80によって使用される。

【0110】リトライRETRY'。この入力はメモリ10がビジーとなっており、転送コントローラ80が再びメモリサイクルを開始すべきことをマルチプロセッサ集積回路100に表示するものである。行ラッチRL'。こ*

STATUS [4:0]

00000
00001
00010
00011
00100
00101
00110
00111
01000
01001
01010
01100
01101
01110
01111
10000
10001
10010
10011
10100
10101
10110
10111
11000
11001
11010
11011
11100
11101
11110
11111

通常の読み出し
通常の書き込み
リフレッシュ
保留
周辺デバイスバケット転送読み出し
周辺デバイスバケット転送書き込み
保留
保留
保留
ブロック書き込み
保留
保留
ロードカラーレジスタ
保留
保留
フレーム0フルシフトレジスタ読み出し
フレーム0フルシフトレジスタ読み出し
フレーム0スプリットシフトレジスタ読み出し
フレーム0スプリットシフトレジスタ読み出し
フレーム1フルシフトレジスタ読み出し
フレーム1フルシフトレジスタ読み出し
フレーム1スプリットシフトレジスタ読み出し
フレーム1スプリットシフトレジスタ読み出し
保留
保留
保留
保留
バケット転送フルシフトレジスタ読み出し
バケット転送フルシフトレジスタ書き込み
保留
保留

表7

【0113】行時間ステータスコードの詳細は次のとおりである。バケット転送、キャッシュミスまたはダイレクト外部アクセスリクエストに対しては通常の読み出しコードが出力される。バケット転送、データキャッシュ書き戻しまたはダイレクト外部アクセスリクエストによって発生される通常の書き込みサイクルに対しては通常の書き込みコードが出力される。トリックルリフレッシュサイクルおよびリフレッシュコントローラからの緊急リフレッシュリクエストにより発生されるバーストリフレッシュサイクルの間にリフレッシュコードが出力される。

【0114】下記に周辺デバイスバケット読み出しおよ

*の出力はアドレスバス上に有効な32ビットアドレスが存在していることを表示するものである。

【0111】ステータスコードSTATUS [4:0]。この出力は現在の転送コントローラのメモリサイクルのタイプと、起点についての詳細な説明を与える。表7に示すように、行アクセスの開始から列アクセスの開始までに、STATUS [4:1]の出力がコード化される。下記のUTIME'と共にこれらをコーディングすることにより、マルチプロセッサ集積回路100のタイミングに発生するメモリタイミングを外部デバイスが発生することができるようになっている。

【0112】

【表7】

アクティビティ

び書き込みについて更に説明する。周辺デバイスのバケット転送の結果として生じるメモリ読み出しサイクルに対しては、周辺デバイスバケット転送読み出しコードが出力される。このことは、後の列アクセス上のメモリから読み出されるデータを、転送を開始させた周辺デバイスによってラッチすべきことを示している。周辺デバイスのバケット転送の結果として生じるメモリ書き込みサイクルに対しては、周辺デバイスバケット転送書き込みコードが出力される。このことは、周辺デバイスがメモリ内に入れるべきデータでバスをドライブできるように、後の列アクセスの間にマルチプロセッサ集積回路100がデータバスを高インピーダンスにすることを意味

している。

【0115】ブロック書き込みは特別なVRAMサイクルである。ブロック書き込みアクセスモードでパケット転送によって発生されたVRAMに対し、ブロック書き込みサイクル中にブロック書き込みコードが出力される。アクセスモードとしてブロック書き込みを指定したパケット転送のカラーレジスタロード部分の間でロードカラーレジスタコードが出力される。このサイクルはシステムVRAMのカラーレジスタにデータを入れるのに使用され、このカラーレジスタデータは部分書き込みサイ

10 クル中にメモリに書き込まれる。
【0116】シリアルレジスタ転送アクセスモードを使用してパケット転送のソースサイクルの間にパケット転送シフトレジスタ読み出しコードが出力される。このサイクルはシステムのVRAM上で読み出しメモリからレジスタへの転送を行うものである。シリアルレジスタ転送アクセスモードを指定したパケット転送の宛て先サイクルの間に出力される。このサイクルシステムVRAM上での書き込みレジスタからメモリへの転送を実行す

る。
【0117】このフレームサイクルはフレームコントローラ90によって必要とされるメモリアクセスである。シフトレジスタの読み出しおよび書き込みは特別シリアルレジスタのVRAMサイクルとなっている。位置読み出し転送コードのうちのフレーム0は、フレームコントローラ90によってリクエストされるそれぞれのフル読み出し転送サイクルの間に出力される。これらサイクルはシステムVRAM上でのフル読み出し転送サイクルを実行し、STATUS [2] 信号はフレーム0のリクエストサイクルでは0であり、フレーム1のリクエストサイ

てリクエストされるフル書き込み転送サイクルの間にフレーム0または1の書き込み転送コードが出力される。これらサイクルは、システムVRAM上でのフル書き込み転送サイクルを実行する。STATUS [2] 信号はフレーム0のリクエストサイクルに対しては0であり、フレーム1のリクエストサイクルに対しては1である。フレームコントローラ90はスプリット読み出し転送サイクルをリクエストするとフレーム0または1のスプリット読み出し転送コードが出力される。これらサイクルはVRAMメモリの行からシリアルレジスタの半分への転送を行う。STATUS [2] 信号はフレーム0のリクエストサイクルに対しては0であり、フレーム1のリクエストサイクルに対しては1である。フレームコントローラ90によってリクエストされるスプリット書き込み転送サイクルの間に、フレーム0または1のスプリット書き込み転送コードが出力される。これらサイクルはVRAMシリアルレジスタの半分からメモリアレイ内へ転送を行う。STATUS [2] 信号はフレーム0のリクエストサイクルに対しては0であり、フレーム1のリクエストサイクルに対しては1である。

20 【0118】別アクセスからのスタートから行アクセスのスタートまでに、表8に示すように、STATUS [4:1] の出力がコード化される。この情報は、転送コントローラ80を介してマルチプロセッサ集積回路100により出力され、メモリサイクルをリクエストするモジュールを示す。これら信号は、システム解析およびデバッグのため発生されるが、外部メモリシステムには不要である。

【0119】

30 【表8】

STATUS [4:0]

アクティビティ

00000	DIGP0	低優先度パケット転送
00001	DIGP0	高優先度パケット転送
00010	DIGP0	キャッシュ
00011	DIGP0	DEA
00100	DIGP1	低優先度パケット転送
00101	DIGP1	高優先度パケット転送
00110	DIGP1	キャッシュ
00111	DIPG1	DEA
01000	DIGP2	低優先度パケット転送
01001	DIGP2	高優先度パケット転送
01010	DIGP2	キャッシュ
01011	DIPG2	DEA
01100	DIGP3	低優先度パケット転送
01101	DIGP3	高優先度パケット転送
01110	DIGP3	キャッシュ
01111	DIPG3	DEA
10000	MP	低優先度パケット転送
10001	MP	高優先度パケット転送
10010	MP	緊急パケット転送インタラプトイネーブル
10011	MP	緊急パケット転送インタラプトディスエーブル
10100		保留
10101		保留
10110	MP	インストラクションキャッシュインタラプトイネーブル
10111	MP	インストラクションキャッシュインタラプトディスエーブル
11000	MP	DEAインタラプトイネーブル
11001	MP	DEAインタラプトディスエーブル
11010	MP	データキャッシュインタラプトイネーブル
11011	MP	データキャッシュインタラプトディスエーブル
11100		フレームメモリ0
11101		フレームメモリ1
11110		リフレッシュ
11111		保留

表8

【0120】転送/出力イネーブル反転TR。転送コントローラからのこの出力は、DRAM出力ドライバおよびVRAMシフトレジスタ転送サイクルをイネーブルする。ユーザータイミング選択反転UTIME。この入力

は転送コントローラが反転RASおよび反転CAS [7:0]のタイミングを変え、ユーザーが自らのメモリタイミングを発生できるようにするものである。この入力はマルチプロセッサ集積回路100が作動するエンディアンモードを決定するようにリセットでも使用される。書き込みイネーブル反転WE。これは通常は書き込みまたは書き込み転送サイクルが生じていることをメモリに表示する出力である。内部メモリテストモードでは、この信号は内部メモリへの書き込みを行うよう、外部デバイスによってドライブされる入力となっている。

【0121】マルチプロセッサ集積回路100により発生される各外部メモリサイクルは、ページモードサイクルを除く期間中の少なくとも5つのマシンステートである。マシンステートは1クロック期間の長さであり、CLKOUTの降下エッジで始まる。各メモリサイクルは2つの部分、すなわちアドレスサブサイクルとデータサブサイクルとを有する。ページモードサイクルは、アクセスが一つのアドレスサブサイクルとマルチデータサブ

サイクルを有するこのフォームの拡張である。

【0122】アドレスサブサイクルは外部メモリサイクルの第1マシンステートで開始し、少なくとも4つのマシンステート長さである。このときにアクセスのためのアドレスおよびステータスコードが出力される。この時間にDRAMおよびVRAMのための行アドレスがラッチされるので、このサイクル部分は、行アドレス時間とも称される。

【0123】アドレスバスA [31:0]は、現在アクセス中の64ビットワードの開始バイトをポイントする32ビットのアドレスを出力する。このアクセスは転送すべきスタートバイトおよびデータ量に応じて1~8バイトのいずれでもよい。このアドレスはマルチプロセッサ集積回路100の外部メモリスペースをデコードするのに使用される。次に外部デコードロジックは多数の信号をマルチプロセッサ集積回路100へ戻し、アドレスシフト選択AS [2:0]によりアクセスされているデバイスのタイプ、列タイミング選択CT [1:0]により速度、ページサイズ選択PS [2:0]によりページサイズ、バスサイズ選択BS [1:0]によりデータバス幅を表示する。この情報はアドレスサブサイクルの長さのみならず、データサブサイクルの長さ、アドレス指

定およびその数を決定するのに使用される。アドレスおよびステータスは RL' または RAS' によりラッチできる。

【0124】アドレスサブサイクルは、実行されるアクセスのタイプが必要とするような4つのマシンステートを越えるマシンステートの整数だけ自動的に延長される。このサブサイクルは更に待機ステートの挿入によっても延長される。このサブサイクルは少なくとも1つのマシンステートの長さであり、アドレスサブサイクルの直後に続く。この時には、DRAMおよびVRAMのための別アドレスが出力され、マルチプロセッサ集積回路100を外部メモリとの間でデータが転送される。このメモリサイクル部分は、列アドレス時間と称される。

【0125】データバスD[63:0]は、マルチプロセッサ集積回路100と外部メモリの間でデータを転送する。これらデータは書き込みサイクルでドライブされるか、または読み出しサイクルでラッチされるかのいずれかである。バス上の有効データの位置はマルチプロセッサ集積回路100のエンディアンモード、転送データ量およびメモリ幅により決定される。

【0126】この時間の間に出力される列アドレスは、32ビットバイトのアドレスのシフトされた変形例である。アドレスバスA[31:0]バス上のアドレスのアライメントはアドレスサブサイクル中にマルチプロセッサ集積回路100へ入力されるアドレスシフト選択AS[2:0]によって決定される。

【0127】データサブサイクルの長さは、通常、アドレスサブサイクル中にマルチプロセッサ集積回路100へ入力される列タイミング選択CT[1:0]により決定されるような1つ、2つまたは3つのマシンステートである。より長いアクセス時間を必要とするデバイスは、アドレスまたはデータサブサイクルのいずれかに待機ステートを挿入できる。

【0128】現在のメモリアクセスは、行時間におけるページサイズ選択PS[2:0]入力に基づく先のアクセスと同一方向で、かつ同一メモリページ内にあるときはいつも、転送コントローラ80はページモードサイクルを使用する。ページモードサイクルは後にマルチデータサイクルが続く一つのアドレスサブサイクルから成り、データは同じメモリページ内に限り隣接している必要はない。

【0129】アドレスおよびデータサブサイクルの双方の間で、マルチプロセッサ集積回路100はSTATUS[4:0]ピン上にステータスコードを出力する。これらステータスコードは実行中の外部サイクルに関する情報を与えるものである。アドレスサブサイクルの行時間の間、STATUS[4:0]ピンは表7に示すようなコードを出力し、実行中のサイクルのタイプを表示する。その行時間ステータスコードは反転 RL または反転 RAS 信号によってラッチでき、メモリバンクデコーデ

ィングを行ったり、特別ハードウェア機能をイネーブルするように外部ロジックによって使用される。データサブサイクルの列時間の間、STATUS[4:0]ピンに関する情報が変化し、そのサイクルおよびそのリクエスト中のプロセッサについての詳細が与えられる。表8にはこれらアクティビティコードが示されている。これらコードはサイクル自体のタイプに関する情報を与えるものでないで、システムのデバッグを容易にするように、主として使用されるものである。

10 【0130】転送コントローラ80が外部メモリと正しく通信できるようにするため、アクセス中のメモリがどのタイプであるかを知る必要がある。これはマルチプロセッサ集積回路100のアドレスシフト選択AS[2:0]、バスサイズ選択BS[1:0]、列時間選択CT[2:0]およびページサイズ選択PS[2:0]入力を用いることにより、行時間でメモリタイプを識別することにより行われる。マルチプロセッサ集積回路100は外部メモリアドレスおよび行時間ステータスコードを出力し、これら入力をサンプリングし、メモリタイプを決定する。これによりアドレスをデコードし、メモリ識別信号を適当なレベルにドライブするための外部ロジック時間が与えられる。選択されたメモリタイプは、次のアドレスサブサイクルまで有効な状態に留まる。

20 【0131】マルチプロセッサ集積回路100はDRAMのみならずSRAMもサポートしているので、アドレスバス上に多重化された行および列アドレスを発生しなければならない。マルチプロセッサ集積回路100は常に行時間にフルの32ビットバイトのアドレスを出力する。列時間で、この回路はバス上のアドレスをシフトしてDRAMによってすでにラッチされている行アドレスに列アドレスを一致させなければならない。アレイサイズはデバイスの行/列アドレスビットの数を決定するので、列アドレスを正しく合わせるにはマルチプロセッサ集積回路100はアクセスするDRAMのアレイサイズに知っている必要がある。これはアドレスシフト選択AS[2:0]入力を使用して行われる選択である。外部論理は行時間においてマルチプロセッサ集積回路100によって出力されるアドレスをデコードし、アドレスシフト選択AS[2:0]入力に3ビットのシフトコードを供給する。転送コントローラ80はこの値をサンプリングし、ラッチし、これを使ってダイナミックメモリに必要なようにアドレスを多重化するか、またはスタティックメモリおよび周辺機器と使用するため、多重化しな

30 40 50 しておくかを決定する。
【0132】表4aおよび4bは、サンプルされたアドレスシフト選択AS[2:0]の値が列時間に出力されるアドレスにどのように影響するかを示している。値が000である場合、各々の後の列アドレスにたいしてシフトされていない32ビットのアドレスが出力される。値が0でない場合、表4aおよび4bが示すように、そ

の後の別アドレスがシフトされる。シフト値は8～14ビットの範囲であるが、このことは64K×N個～256M×N個のレイサイズに対応する8～14個のアドレスピンを備えたダイナミックメモリがサポートされることを意味している。アドレスバスA[2:0]はシフト量に関係なくバイトアドレスビット0～2を常に出力する。これはダイナミックバスのサイジングをサポートするため行われる。

【0133】例えば1M×4個のDRAMが64ビットワイドのコンフィギュレーションでデータバスに接続されていると仮定する。これらメモリは各々10ビットの行および列アドレスを必要とする。マルチプロセッサ集積回路100のアドレスシフト選択AS[2:0]はバイトアドレスを示しており、メモリバンクは64ビット幅であり、このバイトは反転CAS[7:0]ストローブにより制御されるので無視できる。このことは、メモリはビットA[3]で始まる20個の隣接するアドレスビットを必要とする。表4aおよび4bを検討すると、011アドレスシフト選択のAS[1:0]の値は16ビットの多重化されたアドレスを与えることが判る。ビットA[3]の開始アドレスが列時間におけるアドレスピンA[13]に対応しているので、A[13]で開始する10個のマルチプロセッサ集積回路100のアドレスピンにはDRAMアドレスピンが接続されている。

【0134】可能であれば転送コントローラ80はページモードサイクルを実行するので、コントローラは現在アクセス中のメモリのためのページまたは行境界を、いつクロスして新しいページ上の行のアクセスを行うことができるかを知る必要がある。現在のアクセスに対するページサイズは、外部ロジックによりページサイズ選択PS[2:0]ピンに置かれる3ビットの値により表示される。転送コントローラ80は行時間でこれらピンをサンプリングし、これらを使用して変化しているどのアドレスピンがページの変化を表示しているかを決定する。サンプリングされた値は、次の行アクセスまでに転送コントローラ80によって保持される。ページサイズ選択PS[2:0]に表示されたページサイズは、必ずしもアドレスシフト選択AS[2:0]上に表示されたシフト量に対応しているわけではないが、この理由は、多数のバンクをインターリーブすることができからで*

PS[2:0]	ページサイズ	比較されるビット
0 0 0	no page mode	31-0
0 0 1	2 K	31-11
0 1 0	4 K	31-12
0 1 1	8 K	31-13
1 0 0	16 K	31-14
1 0 1	32 K	31-15
1 1 0	64 K	31-16
1 1 1	128 K	31-17

*ある。

【0135】外部メモリアクセスが起きるといつも、転送コントローラ80はアドレスの21個の最高位ビットを内部LASTPAGEレジスタ360に記録する。各々のその後の列アクセスのアドレスは、この値と比較される。これは図9にプログラムされている。ページサイズ選択PS[2:0]に入力された値は、比較中にLASTPAGEレジスタ360の6個の最小位ビットを選択的に無視するのに使用される。この比較において、ページサイズ選択PS[2:0]-1の値に等しいビット番号よりも低いLASTPAGEレジスタは無視される。常に15個の最高位ビットが比較され、常に次のアドレスの11の最小位ビットが無視される。LASTPAGEレジスタ136のイネーブルされたビットと次のメモリアドレスとが一致しない場合、ページが変わり、行アドレスサイクルと共に次のメモリアドレスが始まる。ページサイズ選択PS[2:0]=00であれば、ページモードがディスエーブルされ、別の行アクセスと共にその後のサイクルが開始される。

【0136】例えばアクセスされているメモリが他のバンクとインターリーブされない64ビットのデータバスとして接続されている16個の1M×4個のDRAMから成ると仮定する。各メモリデバイスは2の10乗ビットの行サイズを有する。DRAMのページモードサイクルの間に、単一行内の任意の位置にアクセスできるので、コンフィギュレーションのためのページサイズは2の10乗個の位置/ページ×8バイト/位置=8Kバイト/ページとなる。図9が示すように、これは011のページサイズ選択PS[2:0]に対応する。転送コントローラ80が行時間でページサイズ選択PS[2:0]=011をサンプリングすると、コントローラはページ境界がクロスされたかどうかを判断するため、その後のアクセスのビット31～13をチェックする。これはこのような特殊コンフィギュレーションに対するDRAMの行アドレスビットおよびバンクデコードビットに対応する。図9に、8個の可能なページサイズ選択PS[2:0]のための比較されたアドレスビットおよびページサイズが示されている。

【0137】

【表9】

【0138】LASTPAGEレジスタ360は関連する有効ビット361を有する。この有効ビット361

は、LASTPAGEレジスタに記憶されたデータが有効なものとして処理されたか、または無効なものとしてみなされたかを示す。LASTPAGEレジスタ360はリセット、ホストアクセス、フォールトおよび再試行の後、またはアクセス方向の切り替え、例えば読み出しと書き込みとの切り替えの後に、有効なものとなされる。これにより、その後のアクセス前に常に行アクセスが強制的に行われる。更にLASTPAGEレジスタ360はリフレッシュサイクル、フレームコントローラ90のリクエストしたサイクルおよびパケット転送の発生したシリアルレジスタ転送サイクルの前後で無効とみなされるので、これらサイクルは常にシングルのノンページモードサイクルとして生じる。周辺デバイスのパケット転送の特殊な例では、周辺デバイス転送ステータスコードが出力されるように、常に行アクセスで転送が開始する。新しいステータスコードが出力できるように、転送の終了時にLASTPAGEレジスタ360のデータは無効とみなされる。しかしながら転送中はLASTPAGEレジスタ360は周辺デバイス転送が可能な限りページモードサイクルを利用するように、通常作動する。

【0139】現在のアクセスのためのバスサイズを決定するように、行時間でバスサイズ選択PS[1:0]ピンがサンプリングされる。転送コントローラ80は表5に示すように、8、16、32または64ビットのバスサイズをサポートする。バスサイズをセットすることにより、各列アクセス中に転送コントローラ80が転送できるバイトの最大数が決定される。リクエストされたバイト数がバスサイズを越える場合、転送コントローラ80は転送を完了するように自動的にマルチアクセスを実行する。選択されたバスサイズは、転送にデータバスのどの部分が必要となるかも決定する。64ビットのメモリに対しては、全バスを利用できる。32ビットメモリに対しては、ビッグエンディアンモードでデータバスD[63:32]ピンを使用し、リトルエンディアンモードではデータバスD[31:0]ピンを用いる。ビッグエンディアンモードおよびリトルエンディアンモードではそれぞれ16ビットバスはデータバスD[63:48]ピンおよびデータバスD[15:0]ピンを利用し、8ビットバスはデータバスD[63:56]ピンおよびデータバスD[7:0]ピンを用いる。どのようなバスサイズを使用するにせよ、転送コントローラ80は常にデータをバスの適当な部分に一致させ、適当な反転CASストロブを附勢し、有効なバイトしか転送されないように保証する。

【0140】転送コントローラ80はメモリタイミングの4つの基本的組をサポートしている。これにより待機状態を使用することなく、1列アクセス当たり1、2または3クロックサイクルを選択することができる。すべてのタイプは性質がDRAMに類似しているが、各

々は特定タイプのDRAMまたはSRAMに適している。使用するタイミングは表6に示すように、列タイミング選択CT[1:0]入力によって決定される。現在アクセス中のメモリタイプを決定するため、アドレスをデコードした後に外部デコードロジックはこれらピンを適当なレベルまでドライブする。パイプライン化されていない1サイクル/列タイミングは、極めて高速のアクセス時間を有するデバイスと共に使用されるよう設計されている。パイプライン化された1サイクル/列タイミングは、アクセスがパイプライン化されていることを除けば1サイクル/列タイミングに類似している。1サイクルで1つのアドレスを出力し、次のサイクルでそのアドレスに対応するデータがアクセスされる。このタイミングはパイプライン化されたページモードサイクルをサポートするDRAM/VRAMと共に使用するためのものであるが、同期式SRAMデバイスと共に使用することもできる。パイプライン化されていない2サイクル/列タイミングサイクルは、SRAMおよび高速DRAMデバイスに対して列アクセス時間の2クロックサイクルを発生する。更にこれらタイミングサイクルは、行アドレス時間に1サイクルを加えることも行う。パイプライン化されていない3サイクル/列サイクルは、DRAMおよびその他の低速デバイスに用いるためのものである。このサイクルは列アクセス時間の3クロックサイクルを発生する外に、行アクセス時間に2つのクロックサイクルを加える。更に選択された列タイミングは全ページの間、すなわち次の行アクセスが行われるまで有効なままである。外部デバイスの役割は、発生されたアドレスに基づき、適当な列タイミング選択CT[1:0]を供給することである。

【0141】ブロック書き込みサイクルおよびロードカラーレジスタサイクルの間、バスサイズ選択BS[1:0]ピンは異なる目的に対して使用される。ブロック書き込みは64ビットバスに対してしかサポートされていないので、バスサイズ情報は不要である。その代わりに、アドレス指定されたメモリがサポートするブロック書き込みのタイプを表示するように、バスサイズ選択BS[1:0]が使用される。表23には、これらサイクルの間のBS[1:0]の値が示されている。ブロック書き込みについては後に詳細に説明する。

【0142】図8に示した意図する実施例では、転送コントローラ80の外部メモリアンターフェース302によりマルチプロセッサ集積回路100の外部メモリサイクルが発生される。外部メモリアンターフェース302はメモリアンターフェース信号の変化を制御するよう、一連のステートを発生する複雑なステートマシンを含んでいる。発生されるステートおよびそれらのシーケンスは、実行されるサイクルのタイプ、アクセスされるメモリの列タイミング、次に実行すべきアクセスおよびフォールト等のような外部または内部事象に基づき変わる。

意図する実施例では、マシンステートを含むが、当業者であればリードオンリーメモリに固定プログラムを有するマイクロコントローラまたはプログラマブルマイクロプロセッサによって、これら機能を実行できることは理解できよう。

【0143】図10は、外部メモリインターフェース302のための方法を定めた完全ステート図を示す。多数のステートとステート遷移が存在するが、これらのシーケンスは基本的には、実行されるメモリアクセスに対して選択される列タイミングに依存する。更にステートは2つのグループ、すなわち行時間ステートと列時間ステートとに分解できる。

【0144】行時間ステートは、各メモリアクセスのアドレスサブサイクルすなわち行時間を構成し、外部メモリインターフェース302は新しいページアクセスが開始する時はいつもこれらステートに入る。アドレスされるメモリアクセスが決定されるのは、これらステートの間である。各行アクセスの間には、最低4つの行ステートが生じる。これらは次のように定義される。

【0145】すべてのメモリアクセスに対する開始ステートはr1ステート370である。ステート370中、転送コントローラ80は行アドレスA[31:0]およびサイクルタイプSTATUS[4:0]を出力し、すべての制御信号を不動作ステートにドライブする。

【0146】すべてのメモリアクセスに共通な次のステートは、r2ステート371である。ステート371中では、転送コントローラ80は反転RLハイをアサートし、データ転送方向に従ってDDIN'をドライブする。ステート371中に、転送コントローラ80はアドレスシフト選択AS[2:0]、バスサイズ選択[1:0]、列タイミング選択CT[1:0]、ページサイズ選択PS[2:0]および反転UTIME入力をサンプリングする。

【0147】外部メモリインターフェース302は、ブロック書き込みおよびロード色レジスタを含む読み出しおよび書き込みのために、r3ステート372となる。ステート373中、転送コントローラ80はDBEN'、DSF1、DSF2、反転TRGおよびWE'をアクティブ行時間レベルにドライブし、FAULT'、READYおよびRETRY'入力をサンプリングする。

【0148】外部メモリインターフェース302は、リフレッシュの間に限りrf3ステート373となるステート373中、転送コントローラ80は、すべてのCAS'[7:0]ストロブを附勢し、FAULT'、READYおよびRETRY'入力をサンプリングする。

【0149】外部メモリインターフェース302は、シフトレジスタ転送サイクルに限り、tr3ステート374となる。ステート374は外部的にはr3ステート3

72と同一である。

【0150】外部メモリインターフェース302は、列タイミング選択CT[1:0]=11のときに、3サイクル/列アクセスの間に限り、r4ステート375となる。信号遷移は生ぜず、転送コントローラ80はRETRY'入力をサンプリングする。

【0151】列タイミング選択CT[1]=1のときに、r5ステート376は2および3サイクル/列アクセスと共通である。ステート376中、転送コントローラ80は、RAS'出力を低レベルにドライブし、RETRY'入力をサンプリングする。

【0152】外部メモリインターフェース302はリフレッシュを除くすべてのアクセスの間でr6ステート377となる。ステート377中、転送コントローラ80はRAS'が既に低レベルでなければこれを低レベルにドライブし、DBEN'、DSF1、DSF2、TRG'およびWE'をそれらの適当な行時間レベルにドライブする。転送コントローラ80はREADYおよびRETRY'入力もサンプリングする。ステート377は、多数回繰り返すことができる。

【0153】外部メモリインターフェース302は、リフレッシュサイクルに限り、rf6ステート378となる。ステート378中、転送コントローラ80はRAS'出力を低レベルにドライブし、RETRY'入力をサンプリングする。

【0154】rf7ステート379は列タイミング選択CT[1]=1のとき、2および3サイクル/列リフレッシュと共通である。ステート379中、転送コントローラ80はSTATUS[4:0]に関するプロセッサアクティビティコードを出力し、RETRY'入力をサンプリングする。

【0155】外部メモリインターフェース302は列タイミング選択CT[1:0]=11のとき、3サイクル/列リフレッシュの間に限り、rf8ステート380となる。信号の遷移は生じない。転送コントローラ80はRETRY入力をサンプリングする。

【0156】rf9ステート381は、すべてのリフレッシュに対する最終ステートである。ステート381中、転送コントローラ80はSTATUS[4:0]に関するプロセッサのアクティビティコードを出力し、RETRY'入力をサンプリングする。

【0157】rhizステート382は、高インピーダンスステートである。外部メモリインターフェース302は、外部バスホストリクエスト中にステート382となる。バスリリースを介する画像システムバスの復帰までに外部メモリインターフェース302はステート382を繰り返す。

【0158】列時間ステートは各メモリアクセスのデータサブサイクルまたは行時間を含む。これらステートの間にすべてのデータ転送が行われる。外部メモリインタ

10

20

30

40

50

ーフェース 302 は 2 つの基本シーケンスの列ステート、すなわちパイプライン化された、またはパイプライン化されていない 1 サイクル/列アクセスに対するシーケンスと、2 および 3 サイクル/列アクセスに対するシーケンスをとる。ページモードオペレーションの間、マルチ列アクセスが生じ得るので、外部メモリインターフェース 302 はシングルページモードアクセスが進行する間、何回もこれらステートシーケンスを繰り返すことがある。

【0159】c1、2 ステート 382 は、列タイミング選択 CT [1:0] = 10 の際の 2 サイクル/列アクセスのための第 1 列ステートである。ステート 382 の間、転送コントローラ 80 はアドレスバス A [31:0] 上の列アドレスおよび STATUS [4:0] に関するプロセッサアクティビティコードを出力する。転送コントローラ 80 は、書き込みサイクルであればデータ出力をドライブし、適当な CAS' [7:0] ストロブをアサートする。

【0160】c1、3 ステート 384 は、列タイミング選択 CT [1:0] = 11 の際の 3 サイクル/列アクセスのための第 1 列ステートである。ステート 384 の間、転送コントローラ 80 はアドレスバス A [31:0] 上の列アドレスおよび STATUS [4:0] に関するプロセッサアクティビティコードを出力する。転送コントローラ 80 は、書き込みサイクルであればデータ出力をドライブする。

【0161】c2、3 ステート 385 は、列タイミング選択 CT [1:0] = 11 の際の 3 サイクル/列アクセスのための第 2 列ステートである。ステート 385 の間、転送コントローラ 80 は、適当な CAS' [7:0] ストロブをアサートする。

【0162】c3 ステート 386 は、列タイミング選択 CT [1] = 1 の際の 2 および 3 サイクル/列アクセスのための最終列ステートである。ステート 386 の間、転送コントローラ 80 は、READY 入力をサンプリングし、そのときのサイクルが読み出しサイクルであれば、入力データをラッチする。

【0163】cs1 ステート 387 は、列タイミング選択 CT [1] = 0 の際の 1 サイクル/列読み出しのための開始列ステートである。ステート 387 の間、転送コントローラ 80 は、アドレスバス A [31:0] 上の列アドレス、STATUS [4:0] に関するプロセッサのアクティビティコードを出力し、すべての反転 CAS [7:0] ストロブをアサートする。転送コントローラ 80 は入力データをラッチしない。

【0164】cs2 ステート 388 は第 2 開始列ステートである。外部メモリインターフェース 302 は、列タイミング選択 CT [1:0] = 00 のときに限り、パイプライン化された 1 サイクル/列読み出しのために cs1 ステート 387 の後に、常にステート 388 を挿入す

る。ステート 388 中、転送コントローラ 80 は列アドレスおよび次の列アクセスのプロセッサアクティビティコードを出力する。転送コントローラ 80 はステート 388 中に入力データをラッチしない。

【0165】cp ステート 389 は、列タイミング選択 CT [1] = 0 のとき、すべての 1 サイクル/列アクセスに対する基本的列ステートである。外部メモリインターフェース 302 は、必要な開始ステートの完了後に各ページモード列アクセスのためにステート 389 を繰り返す。ステート 389 中、転送コントローラ 80 はアドレスバス A [31:0] 上の列アドレスおよび STATUS [4:0] 上のプロセッサアクティビティコードを出力し、適当な CAS' [7:0] ストロブをアサートする。転送コントローラ 80 は有効データの書き込みのためにドライブアウトし、有効データの読み出しのためにラッチインする。

【0166】cdr ステート 390 は、列タイミング選択 CT [1] = 0 のときの 1 サイクル/列アクセスのための読み出しドレインステートである。外部メモリインターフェース 302 は、最終 cp ステート 389 の後のステート 390 またはページ変化またはパイプラインバブルのため、またはインターフェースがアイドル状態となったために、他の列アクセスがペンディング中でない場合、cs1 ステート 387 を有する。ステート 390 中、転送コントローラ 80 は反転 CAS [7:0] ストロブをアサートしないが、先の cp ステート 389 または cs1 ステート 387 によりアドレス指定される入力データをラッチする。

【0167】外部メモリインターフェース 302 は、列タイミング選択 CT [1:0] = 00 のとき、パイプライン化された 1 サイクル/列書き込みドレインサイクルのための cdw ステート 391 となる。外部メモリインターフェース 302 は新しいページの開始時に r1 ステート 370 に復帰する直前に、ステート 391 となる。ステート 391 中、転送コントローラ 80 はすべての CAS' [7:0] ストロブをスタートし、最終書き込みされたデータでパイプライン化されたメモリアレイを更新する。

【0168】ci ステート 392 は、アイドル列ステートである。必要なドレインステートの後にそれ以上の列アクセスがペンディング中でなければ、外部メモリインターフェース 302 はステート 392 となる。これはパイプラインバブルによって生じ得るか、またはそれ以上のサイクルがリクエストされておらず、ページ変化が生じないためインターフェースがページモードに留まっているからである。

【0169】図 10 に示したステート移行インディケータは、どの条件によって別のステートへの移行が生じるかを決定する。この場合、所定のステートへの移行が生じるには、多数の条件を満たさなければならない。これ

らステート遷移条件および事象のリストは次の通りである。

【0170】CT=XXr 3ステート372の間にラッチされた表示列タイミング選択CT[1:0]の値に対してステート変化が生じる。

【0171】

r、読み出し-読み出しサイクルが生じる。

w、書き込み-書き込みサイクルが生じる。

r t-読み出し転送サイクルが生じる。

w t-書き込み転送サイクルが生じる。

フォールト-r 3ステート中に低レベルにサンプリングされたFAULT' 入力、メモリアクセスがフォールトであることを表示する。

再試行-r 3ステート372に低レベルにサンプリングされたRETRY' 入力、行時間再試行をリクエストする。

待機-r 3ステート372、r 6ステート377またはc 3ステート386中に低レベルにサンプリングされたREADY入力は、そのときのステートの繰り返しをリクエストする。

スピナー-これは転送コントローラ80のパイプラインがロードできるように、別のr 3スピナー377を加えるため内部で発生されたリクエストである。外部メモリアクセス302は、先のページアクセスからバスサイズが変わると、各2サイクル/列書き込み中に1回、各1サイクル/列各中に2回、1サイクル/列アクセス中に1回、ステート372となる。

新しい列-このときのメモリサイクルは同じメモリページの次の別アクセスである。

ドレイン-このときのメモリサイクルは1サイクル/列読み出しサイクルである。このときの読み出しステートの後には、ペンディングとなっている列アクセスはない。

アイドル-そのときのステートの後にペンディングである列アクセスは無く、すべてのドレインステートは完了している。

新ページ-次のメモリアクセスはページの変更および新しい行アクセスを必要とする。

【0172】転送コントローラ80は内部メモリアクセス301および外部メモリアクセス302の双方のうちにパイプラインを含む。そのときのアクセスがまだ完了していないとき、これらパイプラインは転送コントローラ80が必要とするメモリアクセスを待つことができる。例えばバケットリクエストがオンチップメモリからオフチップメモリに転送中である場合、宛て先サイクルはアクセス当たり2サイクルを必要とすることがある。必要とされるデータがバケット転送FIFOバッファ311にあると仮定すると、アクセス当たり1サイクルでバケット転送FIFOバッファ311からデータを抽出できるので、第1サイクルが完了する前

に、パイプラインに別の宛て先サイクルを挿入できる。

【0173】これらパイプラインは通常ユーザーに対して完全にトランスペアレントである。作動に対するこの効果は、パイプラインがドレイン状態となっているときに見えるだけである。外部メモリアクセス302の外部メモリパイプラインに1サイクルをロードするには、パイプラインに既に含まれている他のサイクルと同じメモリページ内にこのサイクルが入っていなければならない。パイプラインに一旦1つのサイクルを挿入すると、これを除くことはできず、このサイクルを発生しなければならない。従って新しいメモリページへのアクセスがリクエストされると、まず最初にパイプライン内のサイクルを完了しなければならない。転送コントローラ80が外部メモリへのバケット転送を行い、フレームコントローラがメモリアクセスを発生する場合、フレームコントローラサイクルがより高い優先度を有していたとしても、フレームコントローラ90がリクエストしたサイクルを発生する前に、現にパイプライン内にある列サイクルを完了しなければならない。キャッシュおよび外部メモリアクセス、ホストアクセスおよび緊急リフレッシュ前、バケット転送一時中断中、および列時間再試行後にも、パイプラインのドレインを行うことができる。緊急リクエストがペンディングのとき、パイプラインのドレインには最高の優先度が与えられる。

【0174】外部メモリアクセス中、作動サイクルが存在していてもよく、この間にアクティブな列アクセスは生じない。これらの期間は転送コントローラ80の内部パイプライン内のバブルの結果である。転送コントローラ80がこのサイクル中に実行される動作がないときにバブルが生じる。これは転送コントローラ80が書き込みサイクルのための次のアクセスを実行するのに十分なデータを有していないこと、バケット転送FIFOバッファ311がフル状態であり、読み出しサイクルでそれ以上データをロードできないこと、また、端にアクティビティのリクエストはないことを理由とする、クロスバの競合の結果として生じ得る。これら条件のいずれも生じない場合、バブルも発生しない。

【0175】バブルの一例は、転送コントローラ80が内部ソースおよび外部宛て先のバケット転送のサービスをしているときである。転送コントローラ80があるときにソースデータの1バイトにしかアクセスできず、そのときに書き込み宛て先データの8バイトしか書き込みできないように、バケット転送がセットされている場合、転送コントローラ80は8回の内部サイクルごとに1回、宛て先アクセスを行うことができるだけである。この結果、転送コントローラ80は次の列アクセスを実行するのに十分なデータの8バイトを持つのを待ちながら、外部メモリ宛て先サイクル内にバブルを挿入する。挿入されるバブルサイクルの回数は、外部メモリサイクルのタイミングに依存する。

【0176】広範なメモリのタイプおよび組み合わせ、更に周辺機器をサポートするため、マルチプロセッサ集積回路100は外部メモリサイクルの拡張、再試行および終了のための種々の方法を提供する。これらの各々について次の章でより詳細に説明する。

【0177】転送コントローラ80がパイプライン内でペンディング中の列アクセスのすべてを完了すると、メモリアccessの終了の準備ができる。しかしながら新しい行アクセスが必要となるまで終了は行われない。次のメモリアccessまでc iステート392において外部メモリ信号はアクティブ状態のままである。これによりDRAMデバイスはページモードステートの状態に維持できる。次のアドレスが先のメモリアccessと同じメモリページ内に入り、同一方向である場合、行アクセスサイクルは不要である。次のメモリアccessが行アクセスを必要とする場合、そのときのページモードアクセスは終了し、新しい行アクセスが開始する。それ以外の外部バスアクティビティがほとんどない場合、新しい行アクセスの発生を必要とするトリックルリフレッシュサイクルの発生により、最終列アクセスの終了後にすぐにほとんどのメモリサイクルが終了される。

【0178】マルチプロセッサ集積回路100はメモリサイクル時間を延長するように、待機ステートの挿入をサポートする。これはREADY入力を用いることによって行われる。このREADY入力は、CLKOUTの効果エッジで適当な時間にサンプリングされたものであり、READYがハイレベルでサンプリングされた場合、サイクルは通常の態様で続行する。READYが低レベルでサンプリングされた場合、現在のマシンステートが繰り返され、次のCLKOUTで再びREADYがサンプリングされる。このメモリサイクルは転送コントローラ80がハイレベルのREADYをサンプリングするまで、そのときのステートを繰り返すように続く。

【0179】転送コントローラ80は多数の待機ステートによりホールドアップされるメモリアccessを終了させるようなタイムアウトまたはアボート機構を有していない。サービスを待つことができる高い優先度の外部メモリアccessのロックアウトを防止するため、適当な時間に完了できないメモリアccessはフォールトとするか、または再試行すべきである。

【0180】任意のタイプのメモリサイクルに対する行時間の間に、待機ステートを加えることができる。列アクセス当たり2サイクルおよび3サイクルの双方の間で、列時間において待機ステートを挿入することもできる。待機ステートは反転RASの降下の前後の行時間の間に2つの位置で待機ステートを挿入できる。適当なコントローラ80は、まずr 3ステート372またはr f 3ステート373またはt r 3ステート374の開始時に、READY信号をサンプリングできる。これによりこの時間で行アドレスおよび/またはサイクルタイプを

デコードし、反転RASの降下前にアドレス指定されたデバイスが、更なるアクセス時間を必要とするかを判断できる。転送コントローラ80がハイレベルのREADYをサンプリングするまで、r 3ステート372が繰り返される。READYが低レベルでサンプリングされるとき、r 3ステート372が繰り返されるが、反転FAULTおよび反転RETRY入力は再サンプリングされない。待機ステートが挿入される前の第1 r 3ステート372サイクル中にアクセスをフォールトするか、または再試行しなければならない。r 6ステート377中でREADY入力を再びサンプリングする。これはRAS'の降下後に起きるので、必要とし得るデバイスに対し、追加RAS' アクセスタイムを発生できる。READYが低レベルでサンプリングされる場合、このREADYが再びハイレベルでサンプリングされるまでr 6ステート377が繰り返される。

【0181】転送コントローラ80は2および3サイクル/列書き込みおよびバスサイズを変更する1サイクル/列アクセスの間に自動的に付加的r 6ステート377を挿入する。転送コントローラ80はこれらr 6ステート377の各々の間で通常の態様でREADYをサンプリングする。従って、低レベルでサンプリングされるREADYから生じる余分なr 6ステート377は、転送コントローラ80によって既に挿入されたステートに加えられる。

【0182】列アクセス当たり1サイクルの間では、列時間待機ステートはサポートされていないので、これらサイクルの間でr 6ステート377を越えてREADYはサンプリングされない。転送コントローラ80は、列アクセス当たり2および3サイクルに対しc 3ステート386の間でREADYピンをサンプリングする。これは列アドレスおよび反転CASの降下後に生じるので、必要とするデバイスに対し延長されたCASアクセス時間を与えることができる。READYがハイレベルでサンプリングされる場合、c 3ステート386は列アクセスを完了させる。そうでない場合、c 3ステート386が繰り返され、ハイレベルでサンプリングされるまで転送コントローラ80はその後のCLKOUT降下エッジごとにREADYをサンプリングする。

【0183】マルチプロセッサ集積回路100は2つのタイプの再試行機構、すなわち行時間再試行および列時間再試行をサポートしている。行われる再試行のタイプは、マルチプロセッサ集積回路100の反転RETRY入力がアクティブ低レベルとなるメモリアccessのどの点にあるかによって決まる。

【0184】行時間再試行は、現在のアクセスが完了できず、再試行が必要なことを外部ロジックがマルチプロセッサ集積回路100に伝えることができる方法を提供する。この行時間再試行は、r 3ステート372の開始点でまずサンプリングされる際、RETRY' 入力を低

レベルにドライブすることによって発生される。発生RETRYが低レベルでサンプリングされる場合、転送コントローラ80はr3ステート372の終了点で現在のアクセスを終了する。次により高い優先度のリクエストがペンディング中でなければ、再試行されるアクセスは新しいr1ステート370と共にすぐに再スタートする。この場合、再試行されるサイクルは、より高い優先度のリクエストのサービスが完了するまで再スタートされない。

【0185】パケット転送サイクル中に行時間再試行が行われ、同等以上の優先度のパケット転送リクエストがペンディング中であれば、行われている再試行が一時中断され、PTMINによって定義されるインターバルが終了したとみなされる。パケット転送が優先度/ラウンドロビンチェーン内の順番に再び達し、パケット転送が再開されるまで、転送コントローラ80は、この一時中断されたパケット転送を再試行することはない。

【0186】列時間再試行は従来の意味では実際には再試行ではない。むしろこれらは行アクセスリクエストと考えることができる。列再試行が生じると、転送コントローラ80はそのパイプライン内でペンディング中のすべての列アクセスを完了し、行アクセスと共に次のアクセスを開始する。従って再試行がリクエストされた後に多数の列アクセスが行われることがある。行アクセスが再スタートされた後にはこれら行アクセスは繰り返されることはない。r3ステート372の後で反転RETRY入力が低レベルでサンプリングされる時間に列時間の再試行が行われる。転送コントローラ80は各CLKOUT降下エッジでr3ステート372の後の各ステートの終了点で、RETRY'をサンプリングするので、反転RETRYは各サンプリング期間中で有効な高レベルまたは低レベルとなっていなければならない。転送コントローラ80が低レベルの反転RETRYをサンプリングする場合、転送コントローラ80のパイプライン内に現在あるすべての列アクセスが完了するとすぐに、そのときのページモードが終了する。再試行が一旦リクエストされると、反転RETRY入力に入力された値はそれ以上の効果がなくなる。しかしながら転送コントローラ80は残りの列アクセスの各々の間で反転RETRYをサンプリングし続けるので、RETRY'は各サンプリング期間中に高または低レベルに維持されていなければならない。通常のオペレーションは、システムがRETRY'を低レベルにドライブし、現在の行アクセスの終了時までこれを低レベルに維持させるためである。

【0187】外部メモリインターフェース302のパイプラインにロードする方法により、読み出しサイクル中にr3ステート375またはr5ステート376でRETRY'をアサートすることは全く効果がない。RETRY'入力はまだサンプリングされており、有効な高または低レベルになければならない。このため、列時間再

試行が望まれる場合、行アクセスの終了時までRETRY'を低レベルにアサートしなければならない。フレームコントローラ90のVRAMアクセスサイクルおよびリフレッシュサイクルは、列時間によって影響されないが、その理由は、性質上、新しい行アクセスは単一列アクセスの後に続くからである。

【0188】システムエラーのため、システムがメモリアクセスを完了できない場合、システムはメモリサイクルをフォールトすることによりマルチプロセッサ集積回路100に通知できる。これによりメモリサイクルを再試行する前にマスタプロセッサ60はエラーを訂正できる。メモリフォールトは行時間で発生できるだけであり、r3ステート372の開始時にFAULT'入力を低レベルにドライブすることにより開始される。転送コントローラ80はメモリサイクルの他の部分の間で反転フォールトをサンプリングすることはない。フォールト機構はフォールトを生じさせたアクセスのタイプに多少依存して変わる。これについては後により詳細に説明する。異なるタイプのサイクルリクエストに対するメモリフォールトは次のようにサポートされている。

【0189】フレームコントローラ90のサイクル。フレームコントローラ90のサイクルの間にフォールトはサポートされておらず、反転FAULTピンは無視される。リフレッシュサイクル。リフレッシュサイクル中にフォールトはサポートされず、反転FAULTピンは無視される。

【0190】デジタル画像/グラフィックプロセッサキャッシュおよびダイレクトメモリアccessリクエスト。これらサイクルの間にフォールトはサポートされる。リクエスト中のプロセッサはフォールトがクリアされるまで完了されるリクエストは見ない。他のデジタル画像/グラフィックプロセッサ71、72、73、74からのリクエストはサービスされ続ける。

【0191】マスタプロセッサのインストラクションまたはデータキャッシュリクエスト。これらサイクル中にサポートされるフォールトはサポートされる。フォールトされたキャッシュリクエストは即座にキャンセルされ、マスタプロセッサ60はインタラプトされる。その他のキャッシュはペンディング中のリクエストをサービスすることができる。

【0192】マスタプロセッサ60のダイレクトメモリアccessリクエスト。これらサイクル中にはフォールトがサポートされる。外部アクセスリクエストが即座にキャンセルされ、マスタプロセッサ60がインタラプトされる。

【0193】パケット転送。パケット転送なかにはフォールトがサポートされる。このパケット転送は一時中断され、そのステートはリクエスト中のプロセッサのパラメータメモリにセーブされる。他のプロセッサからのパケット転送リクエストはまだサービス可能である。セー

10

20

30

40

50

ブされたパラメータ内には転送コントローラ80の内部
 パケット転送ステートが含まれる。パケット転送が外部
 メモリ間で外部メモリなされる場合、外部メモリ間の転
 送バッファのステートもセーブされる。バッファ自体は
 変わらない。セーブされたパケット転送オプションフ
 ィールド内のパケット転送ステートビットは、フォールト
 がソースへの転送または宛て先への転送で生じたかどう
 かを示すのにセットされる。

【0194】一旦、パラメータがセーブされると、転送
 コントローラ80はマスタプロセッサ60のうちのFL
 TSTSレジスタ内の適当なビットをセットし、どのプ
 ロセッサのパケット転送がフォールトしたかを表示す
 る。下記の表13にはFLTSTSレジスタのコーディ
 ングが示されている。FLTSTSレジスタ内のプロセ
 ッサビットもマスタプロセッサ60に対し、パケット転
 送フォールトインタラプトを発生する。マスタプロセ
 ッサ60はどのプロセッサがフォールトされたパケット転
 送をリクエストしたかを探すため、FLTSTSレジ
 スタを読み出すことができる。一旦プロセッサが識別され
 ば、マスタプロセッサ60は一時中断されているパケ
 ット転送パラメータを検査し、フォールトを生じさせた
 メモリアクセスを決定できる。

【0195】パケット転送中にフォールトが生じたこと
 は、デジタル画像／グラフィックプロセッサ71、7
 2、73または74は知らない。このプロセッサは、パ
 ケット転送がまだ完了していないことを知っているだけ
 である。フォールトを訂正したり、またはリクエスト中
 のデジタル画像／グラフィックプロセッサ71、72、
 73または74に、そのパケット転送リクエストのキャン
 セルを求めるのは、マスタプロセッサ60の役割であ
 る。

【0196】マスタプロセッサ60がフォールトを訂正
 できる場合、FLTSTSレジスタ内のビットをクリア
 することができ、パケット転送リクエストは自動的に再
 送信される。フォールトしたパケット転送は、ラウンド
 ロビン優先法における順番を受けると、その内部ステ
 ートがパラメータメモリ内のセーブされたパラメータから
 レストアされ、転送コントローラ80はフォールトされ
 たアクセスでパケット転送を続ける。

【0197】デジタル画像／グラフィックプロセッサが
 リクエストしたキャッシュサービスまたはダイレクト外
 部メモリアクセスリクエスト中にフォールトが生じた場
 合、リクエスト中のプロセッサのパラメータメモリのキャ
 ッシュフォールトアドレス位置に、フォールトの生じ
 たアドレスがセーブされる。マスタプロセッサ60のFL
 TSTSレジスタにおける適当なビットがセットさ
 れ、インタラプトがマスタプロセッサ60に送られる。
 マスタプロセッサ60がパラメータメモリを検査し、フ
 ォールトしたアドレスを決定できる。マスタプロセッサ
 60がフォールトを訂正できる場合、このプロセッサは

FLTSTSレジスタ内のビットをクリアし、リクエ
 ストは再スケジュールされる。フォールトが訂正できず、
 マスタプロセッサ60がリクエスト中のデジタル画像／
 グラフィックプロセッサにキャッシュミスまたはダイレ
 クト外部アクセスリクエストをアボートすることを求め
 る場合、デジタル画像／グラフィックプロセッサにリセ
 ットリクエストを送らなければならない。これによりそ
 のタスクがアボートされる。

【0198】デジタル画像／グラフィックプロセッサ7
 1、72、73または74は、キャッシュまたはダイレ
 クト外部アクセスリクエスト中にフォールトが生じたこ
 とを知らず、そのリクエストが完了していないことを知
 っているにすぎない。フォールトを訂正したり、リクエ
 スト中のデジタル画像／グラフィックプロセッサをリセ
 ットするのは、マスタプロセッサ60の役割である。

【0199】マスタプロセッサ60がキャッシュフィル
 またはダイレクト外部アクセスサイクルのフォールトを
 リクエストした場合、このリクエストは即座にキャンセ
 ルされ、マスタプロセッサ60にメモリアフォールトイン
 タラプトが送られる。このメモリアフォールトインタラ
 プトはフォールトがデータキャッシュフォールトまたはイ
 ンストラクションキャッシュフォールトであったことを
 表示する。データキャッシュフォールトが生じた場合、
 そのアドレスがセーブされ、データがセーブされる。

【0200】オンチップアドレスへの所定のアクセス
 は、反転FAULT入力と独立したフォールトを生じさ
 せ得る。不法なオンチップアクセスが行われると、これ
 が生じる。試みられたサイクルに対する通常のフォール
 ト機構が適応される。オンチップフォールトとは次のと
 おりである。

【0201】データメモリまたはデジタル画像／グラ
 フィックプロセッサパラメータメモリでない16進数02
 000000以下のアドレスとの間でのデジタル画像／
 グラフィックプロセッサのパケット転送。

【0202】データメモリまたはデジタル画像／グラ
 フィックプロセッサパラメータメモリでない16進数02
 000000以下のアドレスとの間でのデジタル画像／
 グラフィックプロセッサのキャッシュサービス、または
 マスタプロセッサ60のキャッシュサービス、またはダ
 イレクト外部アクセス。データメモリまたはデジタル画
 像／グラフィックプロセッサパラメータメモリでない1
 6進数02000000以下のマスタプロセッサ60の
 パケット転送。

【0203】マルチプロセッサ集積回路100によって
 発生されるメモリタイミングと異なるメモリタイミン
 グの発生を望むユーザーを補助するために、UTIME'
 が発生される。このUTIME'が行時間において低レ
 ベルでサンプリングされると、現在のページの残りに対
 して、反転RASおよびCAS' [7:0]出力のタイ
 ミングが変えられる。列アクセスが開始したことを表示

するように、RAS' 信号が変えられる。従ってRAS' の降下によってユーザーがタイミングを決めたCAS' 信号をトリガできる。実際の列アクセスが開始されたときに、マシンステートのみにあってRAS' をアクティブ低レベルにアサートできるにすぎないので、転送コントローラ80のパイプライン内のバブルを理解することができる。行アドレスと同時に列アドレスストロブCAS' [7:0] が出力される。これにより、これらストロブはどのバイトがアクセスされたかを表示するのに使用でき、外部CAS' の発生が容易となる。RAS' のタイミングは変わっているので、必要な場合には外部ロジックが自らのRAS' タイミングを発生しなければならない。外部から発生されたRAS' は、RL' の降下エッジによってトリガできる。サイクルの開始時点で出力されたステータスコードは、そのときのサイクルの間にてTRG'、WE' 等のエモリタイミングを発生するのに必要なすべての情報を提供することに留意されたい。

【0204】転送コントローラ80の内部メモリインターフェース301は、クロスバー50を介し、オンチップメモリにアクセスする。64ビットの内部データバスは、1サイクルにつき0~8バイトを転送できる。外部メモリインターフェース302は、オフチップメモリおよび周辺機器のすべてにアクセスする。外部メモリインターフェース302は、種々のメモリおよび周辺機器のタイプにインターフェースするのに必要なサイクルおよび制御信号を発生する。64ビットの外部データバスは1サイクルにつき0~8バイトを転送し、8、16、32および64ビットのデータ幅を有する画像システムバスに接続されたデバイスに対し、ダイナミックバスサイズのサポートをする。

【0205】外部メモリインターフェース302は、転送コントローラ80が外部デバイスとバスを共用できるようにする簡単なハンドシェイク機構を含む。このハンドシェイク機構は4つの同期信号を活用する。すなわちホストリクエスト入力HREQ'、ホストアクノリッジ出力HACK' および内部リクエスト出力REQ [1:0] である。

【0206】外部デバイスが画像システムバスの制御を*

* 望む際、ホストリクエスト入力HREQ' は、マルチプロセッサ集積回路100に信号を送る。外部デバイスはバスのオーナーシップを持ちたい場合、ホストリクエスト入力HREQ' を低レベルにドライブする。これは転送コントローラ80が受けることもできる最高優先度のリクエストであり、転送コントローラ80は、最も初期に起こり得る時間にバスのドライブを停止させる。そのときのオペレーションが終了し、転送コントローラ80のパイプラインが空になると、転送コントローラ80のバスオーナーシップが終了する。外部デバイスはバスを望む限り、HREQ' を低レベルアクティブにドライブし続けなければならない。HREQ' がハイレベルで非アクティブとなると、転送コントローラ80はバスを所有し、これをドライブする。HREQ' 入力は内部的にマルチプロセッサ集積回路100の内部ブロックに同期している。

【0207】ホストアクノリッジ出力HACK' は、転送コントローラ80が画像システムバスの制御信号を発生する準備が完了した信号を発生する。転送コントローラ80はアクティブなHREQ' の後に、HACK' を低レベルにドライブし、その信号を高インピーダンスにドライブし、バスを放棄していることを表示する。反転HACKが低レベルアクティブであると、CLKOUTを除くすべての外部メモリインターフェース302のバス出力が、高インピーダンスとされる。内部リクエスト出力REQ [1:0]、HACK' およびフレームコントローラ90の出力は、ドライブされ続ける。外部デバイスは必要に応じて画像システムバスをドライブできる。転送コントローラは、HREQ' が非アクティブとなったことが検出された後に、非同期的に、HACK' をハイレベルで非アクティブにドライブし、バスのドライブを再開する。

【0208】内部リクエスト出力REQ [1:0] は、転送コントローラ80によって受信される最高優先度の内部リクエストの2つのビットコード化を形成する。表3にリクエストコードおよびそれらの関連するサイクルを示す。

【0209】

【表10】

REQ [1:0]	関連内部リクエスト
0 0	低優先度パケット転送 トリクルリフレッシュ アイドル
0 1	高優先度パケット転送
1 0	インストラクションキャッシュサービス データキャッシュサービス
1 1	MP緊急パケット転送 フレームコントローラアクセス 緊急リフレッシュ

表10

【0210】外部ロジックはいつバスを放棄して転送コントローラ80に戻すかを決定するため、内部リクエ

ト出力REQ[1:0]をモニタできる。転送コントローラ80によってホストリクエストには最高の優先度が与えられているので、システムデザイナーは、ホストがバスの所有権を放棄して転送コントローラ80に返すのはどのREQ[1:0]の値にするかを決定することにより、外部デバイスが作動するレベルを決定できる。

【0211】転送コントローラ80は外部メモリシステムが必要とするDRAMリフレッシュサイクルを自動的に発生するためのプログラマブルリフレッシュコントローラを含む。リフレッシュレジスタ313に、このリフレッシュのためのパラメータが記憶される。キャッシュ制御ロジックは、マスタプロセッサ60およびデジタル画像／グラフィックプロセッサ71、72、73および74がリクエストするようなキャッシュフィルおよび書き込みバックを実行するのに必要なアドレスを発生する。フレームコントローラ90のリクエストはVRAM制御論理によって処理される。

【0212】転送コントローラ80はパケット転送を処理するための2つの独立したコントローラを有する。ソースレジスタ321とソースコントローラ322から成るソースマシンは、ソースメモリからデータをフェッチするのに必要なアドレスを発生する。転送コントローラ80にパケット転送リクエストを送る際、このリクエストはソースデータをどのようにアクセスすべきかを指定する多数のパラメータを含む。これらパラメータはソースレジスタ322にロードされ、ソースアドレスを発生するようソース制御ロジック323によって使用される。同様なパラメータの組が宛て先レジスタ341にロードされ、宛て先制御ロジック342によって使用され、宛て先メモリ領域にパケットデータを書き込むのに必要なアドレスを発生する。宛て先レジスタ341と宛て先制御ロジック342の組み合わせにより宛て先マシン340を形成している。ソースマシン320および宛て先マシン340は各々オンチップメモリ10および20とオフチップメモリとの双方のアドレス指定が可能である。

【0213】図11は、ソースマシン320の構造を示す。ソースマシン320の内部構造と宛て先マシン340の構造は同一である。転送コントローラ80の他に対する接続のいくつかが異なっているにすぎない。ソースマシンは次のようなレジスタを含む。CACHEADRレジスタ401は転送方向に応じて内部アドレスまたは外部アドレスのいずれかを含む。このアドレスは、キャッシュサブブロックのアドレスと共に、転送コントローラ80の内部Sポートからロードされる。オンチップアドレスの上部ビットを発生するのに、マルチプレクサを使用できる。

【0214】ABPITCHレジスタ402は、次のラインのための開始アドレスを発生するため、開始アドレスに加算される（または減算される）ピッチを含む。G

TPPIPEレジスタ403は実際には3つの部分に分割されている。第1部分は、ASTARTが宛て先となっている32ビットアドレスである。第2部分はBCOUNTが宛て先となっている16ビットのラインカウントであり、第3部分はACOUNTが宛て先となっている16ビットのバイトカウントである。このレジスタは、関連するレジスタにフェッチしたガイドテーブルを即座に記憶できないとき、ガイドテーブルオペレーション中の一時レジスタとして働く。

【0215】ASTARTレジスタ404は、現在のラインの開始アドレスの記録を記憶し、ASTARTレジスタ404は、ガイドされた転送を実行する際にガイドテーブルアドレス／オフセット入力に対して倍になる。BSTARTレジスタ405は、現在のパッチの開始アドレスの記録を記憶する。BSTARTレジスタ405は、また、オフセットガイドされた転送に対しベースアドレスを含み、デルタガイドされた転送に対しては最終パッチ開始アドレスを含む。

【0216】CRNTADRレジスタ406はパケットサービスの現在のアドレスを記憶する。このアドレスは次のアクセスを実行する場所のアドレスである。次のアドレスを発生するよう転送されるバイト数がこの値に加算されたり、またはこれより減算される。ラインまたはピッチが完了すると、CRNTADRレジスタ406には次のラインまたはパッチの開始アドレスがロードされる。

【0217】BCPITCHレジスタ407は、次のパッチのための開始アドレスを発生するよう、現在のパッチの開始アドレスに加算もしくはこれより減算されるピッチを記憶している。ガイドされた転送を行うとき、このBCPITCHレジスタ407は、ガイドテーブルポインタとして使用される。ACOUNTレジスタ408はライン内のバイト数を記憶している。このバイト数は定数であるか、または可変パッチガイドされた転送のためのガイドテーブル入力からロードされる。ACURRENTレジスタ409はライン内に残っているそのときのバイト数を記憶する。

【0218】BCOUNTはパッチ内のライン数を記憶する。この数は定数であるかまたは可変パッチのガイドされた転送のためのガイドテーブル入力からロードされる。BCURRENTレジスタ411は、パッチ内に残っている1未満のラインの現在の数を記憶する。CCURRENTレジスタ412は、パケット内に残っている1未満のパッチの現在の数を記憶する。ガイドされた転送を行うとき、これはその変わりにガイドテーブル内に残っている入力の数を表示する。

【0219】CBYTESレジスタ413はダイレクトメモリアクセスまたはキャッシュサービスにより転送すべき残っているバイトの現在の数を記憶する。BUFPTRレジスタ414は、ソースマシン320と宛て先マ

シン340によって共用されるレジスタである。BUFPTRレジスタ414は、外部メモリ間でのパケット転送中にバッファとして使用されるパラメータメモリの部分をポイントする。マルチプレクサはオンチップアドレスの上部ビットを発生するのに使用できる。

【0220】FCOUNTレジスタ431(図12に示す)は、パイプライン内に現在あるすべてのインストラクションが完了した後、パケット転送用FIFOバッファ311内にどれだけ多くのバイトが残るかを示す5ビットの値(16~0のみ)を記憶する。

【0221】BUFCOUNTレジスタ441(図13に示す)は、外部メモリ間バッファ内にどれだけ多くのスぺアバイトが残るかを示す8ビットの値(128~0のみ)を記憶する。ソースマシン320の中心は、マシ*

3つの最小位 のアドレス ビット		64ビットバス の前		32ビットバス の前		16ビットバス の前		8ビットバス の前	
前	後	前	後	前	後	前	後	前	後
0	0	0	8	1	4	1	2	1	1
0	0	1	7	2	3	2	1	1	1
0	1	0	6	3	2	3	2	1	1
0	1	1	5	4	1	4	1	2	1
1	0	0	4	5	4	1	2	1	1
1	0	1	3	6	3	2	1	2	1
1	1	0	2	7	2	3	2	1	1
1	1	1	1	8	1	4	1	2	1

表11

【0223】バスが64ビット幅であり、アドレス指定が前方であれば、3つの最小位のアドレスビットの2の補数をとることにより、上記の値を推定する。4番目のアドレスビットは桁上げ用である。次元が後方にアドレス指定される場合、3つの最小位のアドレスビットの値に1を加えるだけでバイト数を計算する。バスが32ビット幅で、アドレス指定が前方であれば、2つの最小位のアドレスビットの2の補数をとることにより、上記の値を推定する。桁上げにより第3ビットとなる。次元が後方にアドレス指定される場合、2つの最小位アドレスビットの値に1を加えるだけで、バイト数を計算できる。これらの場合、計算に通常含まれる次の最高位ビットの影響は無視しなければならない。バスが16ビット幅で前方にアドレス指定される場合、最小位アドレスビットだけの2の補数をとることにより、上記値が推定される。桁上げは第2ビットとなる。次元が後方にアドレス指定される場合、最小位ビットの値に1を加えるだけでバイト数が計算される。これらの場合、通常計算に含まれる次の2つの最高位ビットの影響は無視しなければならない。

【0224】上記の説明は、マシンがそれ自体を現在のバスサイズに一致させるため、転送を望むバイトの数について述べたものである。このオペレーションが行われた後に、連続する転送は自動的に現在のバスサイズによって可能なバイトの最大数となる。このバスサイズを変更する場合、これも自動的に処理される。次にアドレス

*ンが転送したいバイト数を決定するための計算用ハードウェアと方法である。これはバイト計算回路415の数で行われる。これは次元内に残っているバイト数、アドレスの一致、バスサイズおよび次元が前方にアドレス指定されているか、または後方にアドレス指定されているかどうかに応じて決まる。エンディアンはこの計算には重要ではない。次元内に残っているバイト数が8以上であると一時的に仮定すると、表11は異なっているアラインメントのためにマシンが転送を望んでいるバイト数、バスサイズおよびアドレス指定の方向(前方または後方)を示している。

【0222】

【表11】

レジスタはバイト数だけインクリメントまたはデクリメントされ、次のデータの開始点をポイントする。現在の次元カウンタの残りのバイトの新しい数を示すように、同じ数だけデクリメントされる。マシンが転送を望むバイト数が、次元内に残っているバイト数よりも大きくなるまでこのプロセスが続き、残りのバイト数よりも転送を望むバイト数が大きくなった時点まで、実際に転送されるバイト数を低い値にしなければならず、こうして次元の転送を完了する。このような条件は、転送を望むバイト数から残っているバイト数を減算することにより検出され、借りが起こらない場合は次元の最終転送に達し、転送すべきバイト数のためにバイトカウンタを用いる。借り条件に基づくこれら2つの値の選択のためにマルチプレクサが使用される。

【0225】転送が次元内の最終であり、パケットリクエストが実行中であれば、通常の第1次元計算を行う代わりに、次の次元パラメータを計算する。BCURRENTレジスタ411内の値が0であるかどうかにより、BCURRENTレジスタ411またはCCURRENTレジスタ412のいずれかをデクリメントする。いずれの場合にせよ、xCOUNTレジスタから小さい次元のxCURRENTカウンタをロードし、次の第1次元を開始する。

【0226】BCURRENTレジスタ411の値が0となるかどうかに応じて、ASTARTレジスタ404にABPITCHレジスタ402に記憶されていた値を

加算（または減算）するか、またはBSTARTレジスタ404にBCPITCHレジスタ407に記憶されていた値を加算（または減算）することにより、次の次元のためのアドレスも計算される。計算された値は、CRNTADRレジスタ406およびASTARTレジスタ404に書き込まれBCURRENTレジスタ411内の値が0であれば、BSTARTレジスタ405にも書き込まれる。

【0227】第1次元の最終転送が実行され、BCURRENTレジスタ411およびCCURRENTレジスタ412内の値がいずれも0であれば、ソースマシン320はパケットサービスを完了する。ソースマシン320および宛て先マシン340の双方がこの状態に達すると、全パケットサービスが完了する。パケット転送オプションフィールドのソースおよび宛て先更新モードに応じ、ステージで追加次元計算を行うことができ、CRNTADRレジスタ406に記憶されていたデータが下のパケットリクエスト位置にセーブされる。これにより残されている点でのパケット転送の後における再送信が可能となる。

【0228】これらアドレス計算は2つの32ビット加算器420および423で行われる。マルチプレクサ421および422は、加算器420への入力を選択し、同様にマルチプレクサ424および425は加算器423への入力を選択する。加算器420は、CACHEADRレジスタ401、ABPITCHレジスタ402、ASTARTレジスタ404、BSTARTレジスタ405、CRNTADRレジスタ406およびBCPITCHレジスタ407のサービスを行う。加算器423は、ACOUNTレジスタ408、ACURRENTレジスタ409、BCOUNTレジスタ410、BCURRENTレジスタ411、CCURRENTレジスタ412およびCBYTESレジスタ413に対するサービスを行う。

【0229】上記説明は、パケット転送について簡単に触れたものであるが、同様な方法がキャッシュサービスおよびバッファ操作に対しても適用されることに留意されたい。キャッシュサービスの場合、アドレスは常に一致した境界上で開始し、アドレスがキャッシュサブブロックの終了点に達するとサービスが終了するので、バイトカウンタはない。このことは、ソースマシン320がデジタル画像／グラフィックプロセッサのインストラクションキャッシュまたはマスタプロセッサ60のデータまたはインストラクションキャッシュにサービスするかどうかに応じて変わる。

【0230】上記説明は、転送するバイト数の計算を常に実行できることを示唆している。パケット転送用FIFOバッファ311は、マシンがその数のバイトを転送するには、過度に満杯または空状態となり得るので、このことは当てはまらない。したがってパイプライン内に

オペレーションをロードする前にさらに計算を行う必要がある。

【0231】ソースマシン320がパケット転送用FIFOバッファ311内に残されている余裕よりも多いバイトを転送したい場合、待機する必要がある。宛て先マシン340がパケット転送用FIFOバッファ311内にあるバイト数よりも多いバイトを転送したい場合も待機する必要がある。宛て先マシン340が読み出しをするのに、パケット転送用FIFOバッファ311内に十分なバイトをソースマシン320がロードしようとする場合、双方の転送を続けることができる。

【0232】これら計算は図12に示されたロジックで実行される。FCOUNTレジスタ431は既にパイプライン内にあるオペレーションが完了した場合、パケット転送用FIFOバッファ311内に入るデータのバイト数を記録する。加算器434はソースマシン320がパケット転送用FIFOバッファ311に転送したいバイト数にこの値を加える。合計値が16を越える場合、ソースマシン320は続けることはできないが、宛て先マシン320がいくらかのデータを空にすることを待たなければならない。加算器434は、17個以上の出力を発生し、この出力はソースマシン320に供給され、このマシンをストールさせる。現在の値または加算器434の17以上の出力により選択された加算の結果のいずれかであるパケット転送用FIFOバッファ311内の新しいバイト数がマルチプレクサ435を通して減算器436へ送られる。減算器436は宛て先マシン340が転送したいバイト数からパケット転送用FIFOバッファ311内の新しいバイト数を減算する。減算器436が借り信号を発生しなければ、宛て先マシン340は続行できる。減算器426が借り信号を発生すると、この借り信号はマルチプレクサ437を制御して、差または減算前の値を選択してFCOUNTレジスタ431にセーブする。

【0233】キャッシュサービスのためにキャッシュバッファ312を使用するときは、同様な組の計算が必要である。マルチプレクサ433に対するモード信号がFCOUNTレジスタ431が計算のソースとして働くか、またはキャッシュバッファ312に記憶されていたバイト数を記憶する同様なCHCOUNTレジスタ432として働くかを選択する。

【0234】外部メモリ間転送を実行するとき、転送プロセッサ80は、リクエスト中のプロセッサのパラメータメモリの一部をバッファとして使用する。この外部メモリ間バッファのステートおよびその内容は管理が必要である。バッファポインタBUFPTRレジスタ414は、ソースマシン240と宛て先マシン340とによって共用されるが、その理由は、このバッファは1度に一つしかアクセスできないからである。BUFPTRレジスタ414は、ソースマシン320および宛て先マシン

340の通常のロジックを使用して、バッファとの間で転送されたバイト数だけインクリメントされる。BUF PTRレジスタ414は長さが8ビットだけでよい。このアドレスの最高位ビットは、各プロセッサに対して固定されており、リクエスト中のプロセッサに基づき、一定のマルチプレクサから供給される。通常の一致/残留スペースロジックを用いて、転送されたバイト数が計算される。外部メモリ間バッファを空にするソースマシン320は、BUFCOUNTレジスタ441を使用してそのバイトに残っている値を記憶する。外部メモリ間バッファを満たす宛て先マシン340は、FCOUNTレジスタ431を使用してバイトに残っている値を記憶する。

【0235】図13は、バッファカウント演算を示している。BUFCOUNTレジスタ441は、転送方向に応じて外部メモリ間バッファ内のスเปアバイトの数または空にすべき残っているデータバイトの数の8ビットカウントである。ソースマシン320は、外部メモリ間バッファを満たしたとき、パケット転送用FIFOバッファ311にロードするバイト数だけBUFCOUNTレジスタ441をデクリメントする。BUFCOUNTレジスタ441が7以下になるとソースマシン320が停止し、パケット転送用FIFOバッファ311が外部メモリ間バッファ内に空にされるまで、宛て先マシン340の作動が続く。従ってソースマシン320は外部メモリ間バッファ内に入るデータよりも多いデータをフェッチできない。外部メモリ間バッファは、少なくとも120バイトで満杯になる。

【0236】外部メモリ間バッファが満杯になる状態と空になる状態の間で、BUFCOUNTレジスタ441は128から減算し、これを外部メモリ間バッファ内に残っているデータバイトの数にしなければならない。外部メモリ間バッファを空にすると、ソースマシン320はBUFCOUNTレジスタ441が0となるまで、これをデクリメントする。宛て先マシン340はパケット転送用FIFOバッファ311を空にし続け、再び方向を反転させる。外部メモリ間バッファの充填が開始する前に、BUFCOUNTレジスタ441は128にリセットされる。マルチプレクサ442および443、およびフル(全)加算器444は、このような必要な演算を行う。

【0237】パケット転送が一時中断されているとき、BUFCOUNTレジスタ441およびBUF PTRレジスタ414は、他のパケットパラメータがセーブされる。レストレーション時にオペレーションを正しく再開できるように、外部メモリ間バッファを空にするか、または満たす、現在の方向を表示するビットもセーブされる。

【0238】図14aおよび14bは、小エンディアンおよび大エンディアン係数のための、それぞれのポイン

タFD PTRおよびFS PTRとパケット転送用FIFOバッファ311との関係を示している。パケットリクエストはソースおよび宛て先の形状を実質的に任意または従属的にできるので、宛て先に対して常に一致するわけではない。図8において、ソースと宛て先の整列がコンスタントに互いに変化した状態で、0バイトのサイクルと8バイトのサイクルとの間でソースマシン320は記憶し、宛て先マシン340はフェッチが可能である。従ってパケット転送用FIFOバッファ311は16バイトのバイト幅のFIFOであり、このFIFOはソースからの0~8バイトをロードできるのと同時に、宛て先への0~8バイトを空にできる。ソースマルチプレクサおよび整列ロジック330はソースから適当なバイトを抽出し、これらバイトをパケット転送用FIFOバッファ311の隣接するFIFOデータを維持する位置に記憶する。宛て先マルチプレクサおよび整列ロジック350は、パケット転送用FIFOバッファ311からの最も古いバイトを抽出し、宛て先に書き込む前に正しい位置に8バイトワードでこれらバイトを一致させる。ソースマルチプレクサおよび整列ロジック330、並びに宛て先マルチプレクサおよび整列ロジック350は、それらの転送を一致でき、できるだけ長くこの状態に留めることができるようになるとすぐに、64ビットのダブルワードの境界にこれらの転送を一致させる。従ってソースマルチプレクサおよび整列ロジック330並びに宛て先マルチプレクサおよび整列ロジック350は、変動する整列状態をサポートする。

【0239】パケット転送用FIFOバッファ311は、サーキュラバッファとして16ビットを取り扱う。データの開始点を任意のバイト位置とすることができ、宛て先マルチプレクサおよび整列ロジック350がデータを抽出する点は、4ビットのFD PTRレジスタによって表示され、FD PTRレジスタはデータを抽出する同じサイクルで変更され、ポインタを新しいデータの開始点に移動する。宛て先マルチプレクサおよび整列ロジック350は、抽出されたバイトを取り込み、宛て先ダブルワード内の正しい位置へこれらを整列させる。パケット転送用FD PTRバッファ311からの抽出および整列は宛て先マルチプレクサおよび整列ロジック350内の8つのパラレルな16ビットパレルシフトから成る16ポジションバイトローテータを使用した1工程で行われる。周辺のバイトをスワップするように、ガンのようなデータの操作を実行し、次に別の回転を行う。宛て先マルチプレクサおよび整列ロジック350も、宛て先に適当なバイトだけが書き込まれるように、同様にバイトストローブを操作する。

【0240】ソースマルチプレクサおよび整列ロジック330からのパケット転送用FIFOバッファ311へのローディングは、実質的に同様なオペレーションである。ソースマルチプレクサおよび整列ロジック330

10

20

30

40

50

は、4ビットのFSPTTRレジスタによってマークされた最初の空いている位置にそのデータを書き込み、新しい最初の空いた位置を有するFSPTTRレジスタを更新する。パケット転送用FIFOバッファ311がフル状態のとき、FSPTTRレジスタは、一旦宛て先マルチプレクサおよび整列ロジック350があるデータを抽出すれば、空となる最初の位置を実際にポイントする。

【0241】宛て先マシン340は、クロックサイクルのスレーブ位相に関するデータを抽出する。ソースマシン320はクロックサイクルの次のマスタ位相時にロードする。大エンディアンおよび小エンディアン作動モードをサポートするため、いずれの方向にもデータ回転を行う必要がある。データの連続性が常に正しくなるようにパケット転送用FIFOバッファ311の外側から見た場合、小エンディアンモードではデータ回転は時計回り方向にし、大エンディアンモードでは反時計回り方向にする。従って操作マルチプレクサおよび整列ロジック330、宛て先マルチプレクサおよび整列ロジック350、ポインタFSPTTRおよびFDPTTRおよびパケット転送用FIFOバッファ311のレジスタは、双方のエンディアンに合わせるように設計しなければならない。

【0242】図12のFCOUNTレジスタ431は、パケット転送用FIFOバッファ311にどれだけ多くのバイトが保持されているかの表示を記憶する。このFCOUNTレジスタ431は、17以上の値を決して含むことができない。このレジスタはこのレジスタ内に記憶された値がパイプライン内に現在あるインストラクションのすべてが完了した後に、パケット転送用FIFOバイト311内にどれだけ多くのバイトが入るかを表示するように、パイプラインのローディング前にソースマシン320および宛て先マシン340によって操作される。これによって、パケット転送用FIFOバッファ311のブロッキングを生じさせるようなパイプラインへのオペレーションのローディングができなくなる。従って、宛て先マシン340は常に十分なバイトを有し、ソースマシン320は常に十分な余裕を有することとなる。この結果、FDPTTRまたはFSPTTRが互いにオーバーテイクをする危険がなくなる。新しいパケットリクエストを開始すると、FDPTTRおよびFSPTTRが0にリセットされるので、これらの双方は同じ位置をポイントし、FCOUNTレジスタ431は0にリセットされる。これによりパケット転送用FIFOバッファ311の正しい初期化が保証される。

【0243】パケット転送が一時中断されると、FDPTTR、FSPTTRおよびFCOUNTレジスタ431のみならず、パケット転送用FIFOバッファ311レジスタ自身も制御される。これによりパケット転送用FIFOバッファ311の全ステートが保持され、よってこのステートはパケット転送の再開時にレストアできる。

【0244】値で満杯にされたパケット転送は、満杯にしている値をパケット転送用FIFOバッファ311の最下位のダブルワードにロードし、FCOUNTレジスタ431を16にセットし、FDPTTRをエンディアンのための正しいデータ開始点にセットする。小エンディアンに対してはバイト0であり、大エンディアンに対してはバイト7である。値で満杯にされたパケット転送中では、FCOUNTレジスタ431の値は変更されない。従って宛て先マシン340は、満杯になったパケット転送用FIFOバイト311を常時見ているので、この満杯にしている値をフェッチし、整列する。ソースマシン320も満杯にされたパケット転送用FIFOバイト311を常時見ているので、必要に際し、停止する。値で満杯にされたパケット転送を一時中断またはレストアする際には、パケット転送FIFOバッファ311のステートは一時中断の別の形態に対する場合のように、セーブされ、レストアされるだけである。従って値で満杯にされたパケット転送を一時中断するためには特別な考慮は不要である。

【0245】ソースマシン320は、2つのパイプラインを含み、宛て先マシン340と別のパイプラインを共用する。パイプラインは内部メモリアンターフェース301および外部メモリアンターフェース302とインターフェースされ、4つの方向のデータ転送、すなわちオンチップからオンチップへ、オフチップからオンチップへ、オンチップからオフチップへ、オフチップからオフチップへのデータ転送を可能とする。ソースマシン320のアドレスパイプライン427およびバイトパイプライン428は、宛て先マシン340のパイプラインから分離されているが、共通のパイプラインの進行を行うように連動されている。これにより、ステップ状に維持しながら内部メモリアンターフェース310および外部メモリアンターフェース320上での独立した行動が可能となっている。あるインターフェースへの不連続性は、他のインターフェースを停止させることもある。アドレスパイプライン420はソースマシン320によってアクセスされるアドレスを記憶し、バイトパイプライン428はバイトストローブおよび整列情報を記憶する。この情報はどのバイトが対応しており、パケット転送用FIFOバッファ311またはキャッシュバッファ312にどれだけ多くのバイトをロードしたり、これらから除くかを表示している。転送コントローラ80のレジスタの転送にあたり、バイトパケット428はレジスタ番号を記憶する。サービスを受けた情報もバイトパイプラインを下方に移働し、オペレーションがいつ完了したかを表示する。

【0246】ソースマシン320と宛て先マシン340は、リクエストパイプライン429を共用する。リクエストパイプライン429はステータス情報、例えば低優先度パケット転送、ダイレクト外部アクセス等の実行中

のオペレーションのタイプ、およびリクエスト中のプロセッサの表示を記憶する。リクエストパイプライン429は、ステータス事象が生じるときを表示するパラレル制御情報も記憶する。

【0247】一つ以上のアクセスが停止する際には、ときどき内部パイプラインにバブルが生じる。転送コントローラ80は、ガイドされた転送中にガイドテーブルの値をフェッチするのに、これらバブルを使用する。これにより、転送コントローラ80の転送中バンド幅をより効率的に使用できる。宛て先レジスタがフリーでない場合、このようなフェッチされたガイドテーブルの値は、GTPPIPEレジスタ403の関連部分に記憶される。

【0248】転送コントローラ80はバイト配列されたパケット転送のみにおいて、データ配列を行うことができる。ソースマシン320と宛て先マシン340とは独立しているため、これらマシンの各々は0バイトの各サイクルと8バイトの各サイクルとの間でフェッチしたり記憶することができる。このことは、ソースアドレスと宛て先アドレスとの相互の整列はコンスタントに変化し得ることを意味している。変動する整列をサポートするため、転送コントローラ80はパケット転送用FIFOバッファ311とソース配列ロジック332と、宛て先整列ロジック352を含んでいる。パケット転送FIFOバッファ311は、16バイトのFIFOレジスタであり、このレジスタはソースからの0~8バイトのロードと、宛て先への0~8バイトを空にすることを同時に行うことができる。ソースマルチプレクサ331およびソース配列ロジック332は、ソースから適当なバイトを抽出し、これらを先のソースバイトに隣接するパケット転送用FIFOバッファ311に記憶する。宛て先配列ロジック352および宛て先マルチプレクサ351は、パケット転送用FIFOバッファ311から最も古いバッファを抽出し、現在アドレス指定されている宛て先の8バイトのダブルワードで、正しい位置にこれらを配列する。パケット転送配列およびFIFO操作は自動的であり、意図する実施例ではプログラマーに対してトランスペアレントである。

【0249】キャッシュバッファ312にはパケット転送用FIFOバッファ311に対して作動が類似する8バイトのバッファであり、キャッシュおよびダイレクト外部アクセスオペレーション中に使用される。キャッシュの内外への転送は常に8バイト幅である。キャッシュバッファ312には外部メモリバスが60ビット幅よりも狭い場合にデータを配列するのを助ける。キャッシュバッファ312をパケット転送用FIFO場合に311と別個に設けたことにより、現在パケット転送用FIFOバッファ312にあるパケットデータを最初に空にすることなく、パケット転送の途中でより高い優先度のキャッシュおよびダイレクト外部アクセスリクエストをサービスすることが可能となっている。

【0250】転送コントローラ80はオンチップメモリにマッピングされる4つの、ユーザーがアクセス可能なレジスタを有する。これらレジスタはロードおよび記憶インストラクションによりマスタプロセッサ60によってアクセス可能である。これらレジスタはデジタル画像/グラフィックプロセッサ71、72、73、74に対してはアクセスできない。表12はこれらレジスタをリストアップしたものである。

【0251】

【表12】

アドレス	レジスタ
0x01820000	REFCNTL
0x01820004	PTMIN
0x01820008	PTMAX
0x0182000C	FLTSTS

表 12

【0252】REFCNTLレジスタは、システムのDRAMリフレッシュサイクルを制御するのに使用される2つの16ビットの値を含む。16ビットREFRATEフィールド(ビット15~0)は、DRAMリフレッシュサイクルを発生するインターバルを決定する。REFRATEにおける値は、各リフレッシュリクエストの間に生じるマルチプロセッサ集積回路100のクロックサイクル数を示す。REFRATEにおける32(16進数0020)未満の値は、DRAMリフレッシュをデイスエーブルする。REFRATEフィールドはリセット時に32(16進数0020)にセットされる。DRAMリフレッシュ中、16ビットの疑似アドレスが外部アドレスバス上に出力され、リフレッシュバンクデコーディングまたはRAS'だけのリフレッシュとともに使用される。16ビットRPARLDフィールド(ビット31~16)は、リフレッシュサイクル中に出力される最大値を含む。現在のリフレッシュアドレスのトラックを維持するのに、リフレッシュアドレスカウンタが使用される。このカウンタはリフレッシュサイクルが起きるたびにデクリメントされ、このカウンタが0に達すると、RPARLDにおける値が再ロードされる。RPARLDフィールドはリセット時に16進数0FFFFにセットされる。

【0253】パケット転送用最小レジスタPTMIN511およびパケット転送用最大レジスタPTMAX512は、パケット転送の作動時間を制御するようになっている。これらレジスタは、図31に示されており、以下、更に説明する。パケット転送用最小レジスタPTMIN511は、より高い優先度のパケット転送によってパケット転送をインタラプトできる前に、転送コントローラ80によってパケット転送のサービスを行わなければならないクロックサイクルの最小数を記憶する。好ましい実施例では、パケット転送用最小レジスタPTMIN511の24の最小位のビットしか実行されない。よ

り高い位のビットからの読み出しは、常に0に復帰し、これらビットへの書き込みは全く効果がない。リセット時にパケット転送用最小レジスタPTMIN511には16進数10000(64Kサイクル)がロードされる。パケット転送用最大レジスタPTMAX512は、タイムアウト前にパケット転送用最小レジスタPTMIN511により指定される時間の後にパケット転送を続けることができる最大時間を決定する。本実施例では、パケット転送用最大レジスタPTMAX512は24の最小位のビットを実行する。より高い位のビットからの読み出しは0に復帰し、これらビットへの書き込みは何らの効果もない。従ってPTMINサイクルの後により高い優先度のパケット転送によりパケット転送をインタラプトでき、PTMIN+PTMAXサイクルの後に同じ優先度の別のパケット転送を実行するため、このパケ

* ット転送を一時中断できる。より高い優先度のパケット転送のため、あるパケット転送を一時中断できるとき、パケット転送用最大レジスタPTMAX512内に記憶された値は、影響しない。リセット時にPTMAXに16進数の値10000(16Kサイクル)がロードされる。

【0254】パケット転送またはデジタル画像/グラフィックプロセッサ71、72、73、74のインストラクションキャッシュフィルサイクルまたはダイレクト外部アクセスサイクル中にフォールトが生じたことを表示するステータスビットを含む。表13は、FLTSTSレジスタ上の特定のビットの意味を示す。

【0255】

【表13】

FLTSTS ビット	名称	機能
31	DIGPC7	保留予約
30	DIGPC6	保留予約
29	DIGPC5	保留予約
28	DIGPC4	保留予約
27	DIGPC3	DIGP3キャッシュ/DEAフォールト
26	DIGPC2	DIGP2キャッシュ/DEAフォールト
25	DIGPC1	DIGP1キャッシュ/DEAフォールト
24	DIGPC0	DIGP0キャッシュ/DEAフォールト
23	DIGP7	保留予約
22	DIGP6	保留予約
21	DIGP5	保留予約
20	DIGP4	保留予約
19	DIGP3	DIGP3パケット転送フォールト
18	DIGP2	DIGP2パケット転送フォールト
17	DIGP1	DIGP1パケット転送フォールト
16	DIGP0	DIGP0パケット転送フォールト
0	M	MPパケット転送フォールト

表13

【0256】転送コントローラ80は、所定のメモリフォールトの検出に基づき、FLTSTSレジスタの個々のビットをセットする。マスタプロセッサ60のリクエストしたパケット転送中にフォールトが生じると、Mビット(ビット0)は1にセットされる。ビット19~16はデジタル画像/グラフィックプロセッサ71、72、73、74によってリクエストされるパケット転送フォールトを表示するのに使用される。DIGP3(ビット19)は、デジタル画像/グラフィックプロセッサ74のパケット転送フォールトを表示し、DIGP2(ビット18)はデジタル画像/グラフィックプロセッサ73のパケット転送フォールトを表示し、DIGP1(ビット17)はデジタル画像/グラフィックプロセッサ72のパケット転送フォールトを表示し、DIGP0(ビット16)はデジタル画像/グラフィックプロセッサ71のパケット転送フォールトを表示する。8つのデジタル画像/グラフィックプロセッサを含む実施例で、パケット転送フォールトをレポートするのにビット23~20が保留される。対応するデジタル画像/グラフィ

ックプロセッサによりリクエストされるパケット転送中にフォールトが生じると、これらビットの各々が1にセットされる。

【0257】DIGPC3ビット(ビット27)は、キャッシュフィルまたはデジタル画像/グラフィックプロセッサ74によりリクエストされるダイレクト外部アクセスオペレーション中にフォールトが生じると1にセットされ、同様に、DIGPC2ビット(ビット26)における1は、キャッシュフィルまたはデジタル画像/グラフィックプロセッサ73によりリクエストされるダイレクト外部アクセスオペレーション中のフォールトを表示する。DIGPC1ビット(ビット25)における1は、キャッシュフィルまたはデジタル画像/グラフィックプロセッサ72によりリクエストされるダイレクト外部アクセスオペレーション中のフォールトを表示する。DIGPC0ビット(ビット24)における1は、キャッシュフィルまたはデジタル画像/グラフィックプロセッサ71によりリクエストされるダイレクト外部アクセスオペレーション中のフォールトを表示する。8つのデ

デジタル画像／グラフィックプロセッサを使用する実施例では、付加的デジタル画像／グラフィックプロセッサによりリクエストされるキャッシュフィルまたはダイレクト外部アクセスオペレーションを表示するのに、ビット31～28が保留される。

【0258】M、DIGP3～0、またはDIGP3～0ビットのうちのいずれかをセットすると、マスタプロセッサ60がインタラプトされる。セットされているFLTSTSビットをクリアすると、関連するパケット転送、キャッシュフィルまたは外部アクセスが再スケジュール化される。適当なビットに1を書き込むと、FLTSTSビットがクリアされ、1つのビットに0を書き込んでも何の効果もない。

【0259】マスタプロセッサ60のレジスタは、スコアボード化されているので、フォールトを生じたダイレクト外部アクセスリクエストは必ずしもこのプロセッサのインストラクションパイプラインを停止するわけではない。マスタプロセッサ60はデータアクセスの別の試みを行う場合に限り、ダイレクト外部アクセス記憶装置のフォールトにより停止する。フォールトの生じたダイレクト外部アクセスロードは、同じ条件下またはダイレクト外部アクセスによりロードされるレジスタを使用する試みをなす場合に、マスタプロセッサ60を停止させる。デジタル画像／グラフィックプロセッサ71、72、73または74によりリクエストされるフォールトの生じたダイレクト外部アクセスは、リクエストされたダイレクト外部アクセスが完了するまでに、常にリクエスト中のプロセッサを停止させる。

【0260】パケット転送はメモリのうちの2つの領域間でのデータのブロック転送である。ソース(Src)メモリエリアから宛て先(Dst)メモリエリアまで、転送コントローラ80によってデータが転送される。ソースおよび宛て先エリアは、オンチップまたはオフチップメモリのいずれでもよい。パケット転送は、転送コントローラ80への領域としてマスタプロセッサ60またはデジタル画像／グラフィックプロセッサ71、72、73、74のうちの1つによって開始される。転送コントローラ80は、図6および7に示すように、リクエスト待ち行列化および優先度決定ロジック303により、固定された優先度決定方法およびラウンドロビン優先度決定方法を用いてリクエストをサービスする。一つのプロセッサがリクエストを送信すると、このプロセッサはプロセッサの実行を続けることができる。パケット転送はプロセッササイクルを追加することなく、転送コントローラ80によって完了される。パケット転送は、異なる優先度レベルに基づいて送信できるので、より高い優先度の転送は低い方の優先度の転送をインタラプトできる。このようなことが生じると、優先度の低い方の転送は転送コントローラ80によって一時中断され、転送内の現在位置が制御される。より高い優先度の転送が完了

すると、インタラプトされたポイントで一時中断された転送が自動的に再開される。

【0261】パケット転送の理解を容易とするため、パケット転送を説明するのに用いる用語の簡単な定義を順に説明する。ラインとはメモリにおける多数の隣接するバイトのことであり、パッチとは開始アドレスが等距離間隔にしているライングループのことであり、パケットとはパッチの組であり、ピッチとは2つのライン間または2つのパッチの開始点の間のアドレスの差であり、パラメータテーブルとはデータのパケットおよびどのようにソースから宛て先に移動すべきかを記述するパラメータの、8ダブルワード長の組であり、リンクされたリストとはパラメータテーブルの組であり、テーブルの各々はリスト内の次のテーブルをポイントするものである。ガイドテーブルとは、パケット転送における個々のパッチを記述するパラメータのテーブルであり、ソース転送とは、ソースメモリ位置からのデータの転送のことであり、宛て先転送とは、宛て先メモリ位置へのデータの転送のことである。

【0262】各プロセッサは関連するパラメータメモリを有し、転送コントローラがプロセッサからのパケット転送リクエストにサービスする際、転送コントローラ80が使用するため、関連するパラメータメモリ内に多数の位置が別個にセットされている。図15にはこれらエリアが示されている。転送コントローラ80により使用されるパラメータメモリエリアは、転送コントローラ80の使用のみに限定されているわけではない。しかしながら、これら位置に入れられるデータはパケット転送オペレーション中、転送コントローラ80によってオーバーライトされる可能性があることを知っておく必要がある。パケット転送リクエストがアクティブである時、これら位置に書き込みを行ってはならない。その理由は、このような書き込みによってパケット転送またはそのデータが破壊されるからである。

【0263】プロセッサがパケット転送を初期化するためのプロセス、方法またはシーケンスの一例は次のとおりである。対応するパラメータメモリ内にパケット転送パラメータテーブルを作成する。必要であれば、オンチップメモリ内にパケット転送用ガイドテーブルを作成する。ガイドテーブルの性質については後に更に説明する。第1パラメータテーブルの開始点をポイントするように、パラメータメモリ内のリンクされたリストの開始アドレスをセットする。図15は、このリンクされたリストの開始アドレスを記憶すべき対応するパラメータメモリ内の位置を示す。マスタプロセッサ60に対し、このリンクされたリストの開始アドレスは、16進数010100FCから16進数010100FFのアドレスに記憶しなければならない。デジタル画像／グラフィックプロセッサ71、72、73、74に対しては、このようなリンクされたリストの開始アドレスは16進数0

100#0FC~16進数0100#0FF（ここで#はデジタル画像／グラフィックプロセッサの数である）のアドレスに記憶しなければならない。次に適当なパケット転送優先ビットおよびPビットをセットして、転送コントローラ80にリクエストを送る。マスタプロセッサ60のためのPKTRREQ制御レジスタ内およびデジタル画像／グラフィックプロセッサ71、72、73、74のCOMMレジスタ内に、これらビットが位置する。転送コントローラ80は、これらレジスタのステータスを検出し、図6および7に示された優先度に従ってパケット転送リクエストのサービスをする。

【0264】パケット転送リクエストはリンクされたリスト構造のように送られる。リンクされたリストは単にパケット転送パラメータテーブルの集合であり、ここでは各パケット転送は、リスト上の次の入力の入力ポイントに対するポインタを含む。パケット転送はオンチップまたはオフチップメモリ上で作動できるが、パラメータテーブル自体のリンクされたリストはオンチップメモリに記憶しなければならない。各プロセッサはメモリに記憶された多数のリンクされたリストを有することができるが、一時にそのうちの一つしかアクティブになれない。リクエスト中のプロセッサのパラメータメモリ内の専用のリンクされたリストの開始アドレス位置に、アクティブなリンクされたリストの開始点が記憶される。リンクされたリスト内の各入力パラメータテーブルは、リスト上の次の入力の位置に対するポインタを含む。リストの終了部はパケット転送パラメータテーブルのパケット転送オプションフィールド内のストップビットによってマークされているので、リスト内の最終入力は、いずれの位置もポイントできる。

【0265】図16は、簡単なリンクされたリスト構造を示す。このリストは2つのパケット転送パラメータテーブルを含む。リスト開始ポインタは、第1パケット転送パラメータテーブルの開始アドレスを記憶し、第1パケット転送パラメータテーブル内の第1データは、第2の、すなわち本例では最終のパケット転送パラメータテーブルのアドレス内にある。本例は、2つのパケット転送パラメータテーブルしか含まないが、リンクされたリストは、オンチップメモリに合わせるよう、できるだけ長くすることができる。

【0266】あるプロセッサがPビットをセットすることによりパケット転送を送った場合、転送コントローラ80は対応するプロセッサのQビットをセットすることによって応答する。このことは、そのプロセッサのリンクされたリストがパケットコントローラ80内に待ち行列状になっていることを表示する。パケット転送優先レベルに適当なラウンドロビントークンがリクエスト中のプロセッサに達すると、転送コントローラ80はリクエストのサービスをアクティブに開始する。転送コントローラ80がリクエスト中のプロセッサのパラメータメモ

リ内のリンクされたリストの開始位置を読み出す。転送コントローラ80は第1パケット転送パラメータテーブルの内容を、そのソースマシン320および宛て先マシン340内のレジスタに読み出す。次に転送コントローラ80はソースマシン320および宛て先マシン340を使用して、パラメータテーブル内に表示されたようにデータを転送する。パケット転送が完了すると、転送コントローラ80はリンクされたリスト内の次のアドレスと共に、パラメータメモリ内のリンクされたリストの開始アドレスを更新する。このアドレスは、ちょうど完了したパケット転送パラメータテーブルの第1入力から読み出される。転送コントローラは次に、リンクされたリスト上の最後の入力に達するまで、この手順を繰り返す。

【0267】本明細書に述べたパケット転送デバイス、プロセスおよび方法は、データの移動のフレキシビリティを最大にできるよう、多数の異なるフォーマットおよびオプションを提供するものである。これらフォーマットを検討する際、パケット転送のソース転送と宛て先転送とは互いに独立していることに留意することが重要である。これにより読み出し時のフォーマットと完全に異なるフォーマットを利用してパケットデータを書き込むことができ、これにより任意の数のスプレッド機能またはマージ機能を自動的に達成できる。これら2つの基本的パケット転送フォーマットは、次元が決定され、ガイドされている。これらフォーマットはソース転送またはソース転送を記述しているかまたは宛て先転送を記述しているかに応じて、どのようにデータを読み出すか、または書き込むかを決定できる。ソース転送と宛て先転送に対して異なるフォーマットを規定することが可能であることに留意されたい。

【0268】次元の定められた転送は、最もシンプルなタイプであるが、最も硬直したタイプでもある。このような次元の定められた転送はデータバイトの簡単な隣接するリニアシーケンスでもよいし、または多数のかかる領域から構成できるソースまたは宛て先を記述する。アドレス指定機構は三次元までのアレイの指定を可能とする。この機構は、単一パケット転送による多数の二次元パッチの転送を可能とする。第1またはA次元に沿うデータは、常に1バイトだけ離れている。第2すなわちB次元および第3すなわちC次元に沿う間隔は任意であるが、パッチ全体に対して固定されている。宛て先次元が完了すると、転送が完了する。

【0269】図17は、次元の定められた転送がどのようにソースメモリまたは宛て先メモリにアクセスするかを例示している。この図は3ラインの2パッチから成るパケットを示しており、各パッチは512個の隣接する8ビットピクセルから成る。例えば2つのデジタル画像／グラフィックプロセッサ71、72、73、74が3×3の畳み込み（各々はラインのパッチの1つに作用

する)を実行しようとしている場合、これが必要となり得る。第1パッチ(PQR)は、デジタル画像/グラフィックプロセッサ71と関連するデータメモリ22に転送すべきデータを示し、第2パッチ(STU)データはデジタル画像/グラフィックプロセッサ72に関連するデータメモリ27に転送すべきデータを示す。本例はソースメモリエリアから転送コントローラ80がデータをどのように読み出すべきかを指定しているので、ソース転送を示していることになる。データパケットは、次のパラメータ、すなわちAカウント、Bカウント、Cカウ
10ント、開始アドレス、BピッチおよびCピッチによって特定される。Aカウントとは、第1次元内の隣接するデータバイトの数であり、図17の例ではこの数は512である。Bカウントとは、パッチを形成する工程の数または1未満のライン数である。図12はBカウントが2である場合を示す。Cカウントは1未満のパケットを形成するためのパッチ工程の数である。図17はCカウント数が1である場合を示す。開始アドレスとは、パケットの開始点のリニアアドレスである。このアドレスはP
20と表示されるバイトのアドレスである。Bピッチとは第2次元のリニアピッチのことである。すなわちバイトPとQとのアドレスの差またはバイトQとRとのアドレスの差である。Cピッチとは第3次元のリニアピッチ、すなわちバイトPとSとのアドレスの差のことである。ソース転送および宛て先転送のいずれも、このように定義できるが、ソースと宛て先の形状を全く異なるようにできるように、パラメータは独立したものである。

【0270】次元の定められた転送のすべての次元をアクティブにする必要はない。Bカウントおよび/またはCカウントを0にセットすることにより、転送を個々の
30バイト、ピクセルの倍数バイト、ラインまたはパッチに限定できる。0のAカウント値は転送されるデータを0にするので、エラーを発生し得る。

【0271】ガイド転送とは次元アドレスのシーケンスをパケット転送パラメータ内の値のみから計算するのではなく、オンチップメモリ内のテーブルからガイドする転送のことである。これらのオペレーションは次元の定められた転送よりも複雑であるが、よりフレキシブルである。ガイド転送には2つのクラス、すなわち固定パ
40ッチ転送と可変パッチ転送とがある。固定パッチガイド転送は次元の定められた転送の場合のようにパケット転送パラメータ内に記述された第1次元および第2次元を有するが、第3次元はオンチップガイドテーブル内のエントリーからガイドされる。可変パッチガイドテーブルではガイドテーブルは各パッチに対しA次元およびB次元のサイズも決定する。いずれの形態でも、2つの1次元は次元の定められたテーブルと同じようにアクティブである。従つて各ガイドテーブルエントリーが2つの第1次元のサイズに従つて個々のバイト、倍数バイトの個々のピクセル、ラインまたは2次元パッチを移動できる。
50

これによりルックアップテーブルによるライン描きまたはデータ処理の際に生じるような多数のイレギュラーなオペレーションが可能となる。

【0272】特に表示しない限り、ガイド転送についての次の記載における情報は、ソース転送および宛て先転送の双方にも同じように当てはまる。ガイドテーブルは単なるエントリーのブロックであり、これは転送のタイプに応じて32ビットまたは64ビットとなり得る。このガイドテーブルは、オンチップメモリ内に位置してい
なければならぬ。ガイドテーブルは固定パッチ転送に対しては、32ビットワードのアドレスに整列していなければならず、可変パッチ転送に対しては64ビットの
20ダブルワードアドレスに整列していなければならない。ガイドテーブルが含むガイドテーブルの開始アドレスおよびエントリーの数は、対応するパケット転送パラメータテーブル内に表示される。各ガイドテーブルエントリーはパケット転送内の2次元パッチに対応しており、転送コントローラ80がパケット転送サービスをする際、このコントローラはパケット転送内の次のパッチの処理をするのに必要のように、ガイドテーブルエントリーを一つずつフェッチする。ガイドテーブルエントリーの詳細については、下記のガイド転送の詳細な説明で述べ
る。

【0273】固定パッチガイド転送は、32ビットエントリーを含むオンチップガイドテーブルを使用する。このテーブルは、ワード整列されていなければならないので、テーブルのエントリーアドレスの最小位の2つのビ
ットは00でなければならない。各エントリーは3次元の転送のためのアドレスを計算するのに使用される情報を含む。固定パッチ転送は3つのタイプがある。すなわ
30ち固定パッチデルタガイド転送、固定パッチオフセット転送および固定パッチオフセットガイドルックアップテーブル転送がある。これらタイプについては、下記により詳細に説明する。

【0274】固定パッチデルタガイド転送に対し、ガイドテーブルは現在のパッチの開始アドレスを形成するのに、先の2次元パッチの開始アドレスに追加すべき32
ビットのデルタ値を含む。パッチサイズはAカウントおよびBカウントパケット転送パラメータによって固定さ
40れ、定義される。

【0275】図18に、固定パッチデルタガイドパケット転送のプロセスおよび方法の一例が示されている。ここで、第1パッチすなわちパッチAの開始アドレスを形成するため、パケット転送パラメータ内に示された開始
アドレスに値デルタAが加算される。この開始アドレスは最終パッチ開始レジスタ403に記憶され、パッチBの開始アドレスを形成するのに最終パッチ開始点403
に記憶されたパッチAの開始アドレスに、デルタBが加算され、同様に、その後同じような加算が行われる。こ
50こで加算値は次のパッチの開始アドレスのベースを形成

するよう、最終パッチ開始レジスタ403に常時記憶されることに留意されたい。図18に示すように、テーブルポインタ501は、ガイドテーブル502内の次のエントリーをポイントするように、各パッチの後に4バイトだけインクリメントされる。多数のエントリーカウンタ503には、最初パケット転送パラメータテーブルからのエントリー値の数がロードされており、エントリーカウンタ503の数は各パッチの後で一つだけデクリメントされる。エントリー各503の数が0に達するとパケット転送が終了する。

【0276】固定パッチオフセットガイドパケット転送は、ガイドテーブル502を使用し、このテーブル502は各パッチの開始アドレスを形成するため、パケット転送パラメータ内に示されるベースアドレスに加算すべき32ビットの値を含む。パッチサイズはAカウントおよびCカウントパケット転送パラメータにより固定され、定義される。

【0277】図19は固定パッチオフセットガイドパケット転送のためのアドレス指定機構プロセスおよび方法を示す。ガイドテーブル502における第1エントリーからの値デルタAは、パッチAの開始アドレスを形成するため、パケット転送パラメータテーブルに特定され、ベースアドレスレジスタ505に記憶されたベースアドレスに加算される。次に、パッチBの開始アドレスを形成するよう、ベースアドレスレジスタ505内に記憶されたアドレスにデルタBが加算され、ガイドテーブル502内の最終エントリーまでに同様な加算が行われる。ここで、パケット転送パラメータテーブル内に特定されたベースアドレスが0になると、ガイドテーブル502は絶対アドレスを指定する。デルタガイド転送の場合と同じように、テーブルポインタ501は現在のガイドテーブルエントリーをポイントし、エントリーカウンタ503の数は実行するパッチの数をトラッキングする。

【0278】ガイドテーブル502は、固定パッチオフセットガイドルックアップテーブルパケット転送に対し、32ビットのオフセット値を含む。この値はオフセットレジスタ506内で0フィルの状態で0、1、2または3ビットだけ左にシフトしベースアドレスレジスタ505内に記憶されたアドレスに加算されるべきものである。ベースアドレスレジスタ505に記憶されたベースアドレスは、パケット転送パラメータに示されており、これによりルックアップテーブルのデータサイズと独立したルックアップテーブルのオペレーションのために転送を利用できる。固定パッチオフセットガイドルックアップテーブルフォーマットは、ソース転送に対して使用できるだけである。シフト量はパケット転送パラメータのAカウントフィールドのビット1〜3のうちの最も左側のビットの位置によって表示されている。ビット3のうちの1は、左にシフトされた3つの場所を示し、ビット2は2つの場所を示し、ビット1は1つの場所を

示す。ビット1〜3の中に1がない場合は、0シフトを示す。このような左シフトにより、8、16、32および64ビットのサポートが可能となる。ルックアップテーブル転送のためのパッチサイズは、固定された1次元であり、1、2、4または8バイトにセットされる。

【0279】図20は、固定パッチオフセットガイドルックアップテーブルパケット転送のためのアドレス計算の一例を示す。ここでオフセットAはAカウントフィールド内の値によって示されるようなルックアップテーブルのデータサイズに従って、0、1、2または3ビットだけ左にシフトされる。オフセットレジスタ505内のこのようなシフトされたアドレスは、パッチAの開始アドレスを形成するため、ベースアドレスレジスタ505内に記憶されたベースアドレスに加算される。次に、オフセットB値がシフトされ、ベースアドレスレジスタ505内のベースアドレスに加算され、パッチBの開始アドレスを形成し、同様な操作が次々に行われる。ガイドテーブル502からオフセット値がロードされる際にシフトが行われる。先の実施例で述べたように、テーブルポインタ501およびエントリーレジスタ503の数は、ガイドテーブル502内の現在位置およびパッチの数のトラッキングを維持する。

【0280】可変パッチガイド転送は、パケット転送パラメータ内でなく、ガイドテーブル内のすべてのパッチサイズ情報を特定する。これによりパケット転送内の各パッチが異なる次元を有することができる。転送はデルタガイドまたはオフセットガイドのいずれでもよい。ガイドテーブルは可変ガイドパッチ転送に対し64ビットのダブルワードエントリーから成る。図21にガイドテーブル用の小エンディアンフォーマットが示されている。ダブルワードの下方の半分は第1の2つの次元に対するAカウント値およびBカウント値を含む。上方の32ビットは第3次元のアドレスを計算するのに使用される値を含む。図22は、大エンディアンガイドテーブルフォーマットを示し、このフォーマットでは、上方の32ビットがAカウント値およびBカウント値を含み、下方の32ビットがオフセットまたはデルタを含む。

【0281】32ビットの書き込みを用いることにより、エンディアンフォーマットから独立したガイドテーブルを作成するためのソフトウェアが適当に与えられる。AおよびBカウントはワード1のアドレスに現れ、オフセット/デルタアドレスはワード0のアドレスに現れる。ここでワード1はワード0よりも大きいアドレスの4バイトにある。転送コントローラ80は64ビットのアクセス中に一時に双方の32ビットワードをアクセスし、そのとき選択されているエンディアンモードに従って正しい内部オペレーションをするのに必要なワード順序を調節する。可変パッチ転送のためのガイドテーブルエントリーはダブルワードに整列すべきである。すなわちバイトアドレスの3つの最小位ビットを000とし

なければならない。

【0282】図23は、可変パッチのデルタガイドパケット転送の一例を示す。可変パッチデルタガイドパケット転送に対し、ガイドテーブルエントリーの第1ワードは、最後のパッチ開始レジスタ504内に記憶された先のパッチの開始アドレスに加算すべき32ビットのデルタ量を含む。パケット転送パラメータには第1パッチのための開始アドレスが示されている。パッチサイズは可変であり、各ガイドテーブルエントリーの第2ワード内に指定される。ここで、パッチWの開始アドレスを形成するのに、パケット転送パラメータ内に特定された開始アドレスにデルタWが加算される。Aカウントは第1次元のバイトの数の長さを決定し、Bカウントは第2次元のサイズ（これはライン-1の数である）を決定する。次にパッチXの開始アドレスを発生するよう、最終パッチ開始レジスタ504内に記憶されたパッチW開始アドレスにデルタXが加算される。先に固定パッチ転送の場合に述べたように、テーブルポインタ501およびエントリーカウンタ503の数は、ガイドテーブル502内の位置のトラッキングを維持する。各ガイドテーブルエントリーは、64ビットすなわち8バイトであるので、テーブルポインタ501は、8バイトよりも大きいアドレスをポイントするようにインクリメントされる。

【0283】図24には、可変パッチオフセットガイドパケット転送の一例が示されている。可変パッチオフセットガイドパケット転送のための各ガイドテーブルエントリーの1ワードは、32ビットオフセット値を含む。この32ビットのオフセット値は、各パッチの開始アドレスを計算するよう、ベースアドレスレジスタ505内に記憶されたベースアドレスに加算される。このベースアドレスはパケット転送パラメータ内で特定される。パッチサイズは可変であり、各ガイドテーブルエントリーの他方の半分内に特定される。パッチWの開始アドレスを発生するよう、ベースアドレスレジスタ505内に記憶されているベースアドレスに、アドレスWの値が加算される。Aカウント値およびBカウント値はパッチのサイズを決定する。パッチXのための開始アドレスを得るよう、ベースアドレスレジスタ505内に記憶された下のベースアドレスにオフセットXが加算される。テーブルポインタ501およびエントリーカウンタ503の数は、ガイドテーブル502内の位置のトラッキングを維持しながらテーブルポインタ501が8だけインクリメントされる。

【0284】ルックアップテーブルパケット転送に類似する、値でフィルする（fill-with-value）パケット転送は、ソース転送に対して指定できるにすぎない。この値でフィルするパケット転送は実際にはソースメモリからデータを転送するのではなく、むしろパケット転送パラメータ内にソースの値を特定するようになっている。2つの32ビットフィールド、最大位の

フィル値のワードおよび最小位のフィル値のワードが、宛て先メモリを満たすのに使用される64ビットの値を特定する。フィルパターンが64ビット未満であれば、最小値のフィル値のワードおよび最大位のフィル値のワードにわたって、このパターンを再現しなければならない。フィル値では整列オペレーションが実行されず、宛て先ダブルワードに書き込まれるバイトはフィル値のダブルワードからの対応するバイトとなっている。値でフィルするパケット転送に対して、ソース開始アドレスまたはソース次元カウントは特定されない。パケット転送のサイズは宛て先転送パラメータによって決定される。

【0285】種々の転送フォーマットによりソース転送と宛て先転送の多数の組み合わせが可能となっている。これら組み合わせを特定するため、パケット転送パラメータテーブル内のパラメータのフォーマットは、必要なソース転送および宛て先転送のタイプに応じて変わる。パラメータテーブルの内容の説明を補助するため、図25～29に種々の可能なパラメータの例が示されている。

【0286】パケット転送パラメータテーブルは、オンチップメモリすなわちマスタプロセッサ60のパラメータメモリ15、デジタル画像/グラフィックプロセッサ71、72、73、74のパラメータメモリ25、30、35または40、もしくは、デジタル画像/グラフィックプロセッサ71、72、73、74のデータメモリ22、23、24、27、28、29、32、33、34、37、38または39内にあることが好ましい。パケット転送パラメータテーブルは、整列された64バイトでなければならない。すなわち6つの最小位アドレスビットが000000となる。どのメモリがこのテーブルを含むかについては、制限はない。従ってマスタプロセッサ60はデジタル画像/グラフィックプロセッサ71に対応するパラメータメモリ25にあるパラメータテーブルを使用することができる。リクエスト中のプロセッサはパケット転送リクエストを送る前に自己のパラメータメモリ内のリンクされたリストの開始アドレス位置に適当な開始アドレスを入れるだけである。

【0287】図25～28に関連して、次の章はパケット転送パラメータ内の種々のフィールドについて述べる。PTによって表示されるパケット転送パラメータの開始アドレスに対する各フィールドのアドレスは、フィールドが有効な転送タイプと同じように示される。多くの場合、2つの同じフィールドがあり、1つはソース転送用であり、他方は宛て先転送用である。これらの場合、1つのコンテキストでフィールドの記述が示される。あるフィールドは、転送のタイプに応じて使用しない。これらのケースではフィールドはプログラムしないままにできる。表14は可能なオプションのすべてをリストアップしたものである。以下、これらのオプションについて詳細に説明する。

【0288】

* * 【表14】

オペレーション	ソース	宛て先
次元の定められた転送	yes	yes
固定パッチデルタガイド転送	yes	yes
固定パッチオフセットガイド転送	yes	yes
固定パッチオフセットガイドLUT	yes	no
可変パッチデルタガイド転送	yes	yes
可変パッチオフセットガイド転送	yes	yes
値で満たす転送	yes	no
トランスベアレントの次元の定められた転送	yes	no
トランスベアレント固定パッチデルタガイド転送	yes	no
トランスベアレント固定パッチオフセットガイド転送	yes	no
トランスベアレント可変パッチデルタガイド転送	yes	no
トランスベアレント可変パッチオフセットガイド転送	yes	no
ブロック書き込み	no	yes

表14

【0289】パケット転送パラメータ内の第1データワードは、次のパラメータテーブルの次のパケット転送開始アドレスである。このワードはすべてのパケット転送タイプに存在し、パケット転送用のリンクされたリスト上の次のエントリーの開始点に対する32ビットのポインタである。このワードは64バイトの整列したオンチップアドレスをポイントするので、このアドレスの6つの最小位のビットは000000とならなければならない。パケット転送用オプションフィールド内の停止ビットは、リンクされたリストを終了させるのに使用されるので、リンクされたリストの最終エントリーは、次の特別なアドレスエントリーを必要としない。パケット転送が成功裏に終了するときはいつも、リクエスト中のプロセッサのパラメータメモリ内のリンクされたリストの開始アドレス位置に、次のパケット転送エントリーアドレスワード内の値が書き込まれる。これによりポインタは次のパケット転送に自動的に進む。このポインタは、リンクされたリストに実行を休止するのに停止ビットが使用される場合に、このポインタがリンクされたリストがイネーブルされた際にリンクされたリスト上の次のパケット転送をポイントするように、リンクされたリスト上の最終パケットが完了した後にも、更新される。

【0290】第2ワードは、パケット転送オプションフィールドとなっている。これはPT+4に位置する。このパケット転送オプションフィールドは、データを転送する方法で種々のオプションを指定するようにすべてのパケット転送タイプで使用される。以下、このパケット転送オプションフィールドについて詳細に説明する。

【0291】次元が定められたパケット転送およびデルタガイドパケット転送で使用される第3および第4データワードは、ソースおよび宛て先開始アドレスとなっている。PT+8にはソース開始アドレスがあり、PT+12には宛て先アドレスがある。これらフィールドは次元の定められたパケット転送およびデルタガイドパケット転送で使用される。次元の定められたパケット転送に対し、各々はソースまたは宛て先のための開始バイトアドレスを表示する32ビットのワードであり、デルタガ

イド転送に対し、各々はソースまたは宛て先転送のために第1デルタオフセット値を加算する開始アドレスを示す。

【0292】オフセットガイドパケット転送で使用される第3および第4データワードは、PT+8にはソース開始アドレスがあり、PT+12には宛て先アドレスがある。これらフィールドは、オフセットガイド転送モードを使用する際に、ソースおよび宛て先開始アドレスの場所をとる32ビットのオフセット値である。

【0293】第5および第6データワードは、ソースおよび宛て先のAおよびBカウントであり、PT+16バイトにおける第5データワードはビット15~0にソース用のAカウントを有し、ビット31~16にソース用のBカウントを有する。同様に、PT+20における第6データワードは、ビット15~0に宛て先用Aカウントを有し、ビット31~16に宛て先用Bカウントを有する。このAカウントの16ビットフィールドは、次元の決められた、すなわち固定パッチ転送のためのソースまたは宛て先の対応する第1次元において、転送すべきバイトの数を指定する。可変パッチガイド転送に関しては、Aカウントフィールドは使用されない。更に値でフィルされたパケット転送に関してはソースAカウントは使用されない。Bカウントの16ビットフィールドは、次元の定められた、または固定されたパッチ転送用のソースまたは宛て先の対応する第2の次元で発生すべきステップ数を特定する。これはライン-1の数に等しい。従って0の値は第2の次元をディスエーブルし、その結果パッチ当たり1本のラインしか転送しない。ソースBカウントは値でフィルされたパケット転送中には使用されない。

【0294】次元の定められたパケット転送内のPT+24およびPT+28に位置する第7および第8データワードは、それぞれソースCカウントおよび宛て先Cカウントである。これら32ビットのフィールドはソースまたは宛て先の第3次元におけるパッチ工程数を特定する。従って0の値は第3次元をディスエーブルし、一つのパッチしか転送しない。フィル転送に対してはソース

Cカウントフィールドを使用されない。

【0295】デルタまたはオフセットガイド packets 転送における第7および第8データワードはエントリーの数である。これらの32ビットフィールドにおける値は、それぞれソースおよび宛て先のためのガイドテーブルのエントリーの数を指定する。従ってこれらフィールドは、転送される情報のパッチ数を表示する。値が0であればデータ転送は行われない。

【0296】PT+32およびPT+36には第9および第10データワードがある。次元の定められた固定パッチ packets 転送では、これら32ビットのフィールドはソースおよび宛て先の第2次元のピッチを特定する。このピッチの値はソースまたは宛て先ラインの開始アドレスに加算され、次のラインの開始アドレスを得る。対応するBカウントフィールドにおける値は、0であるが、このフィールドはプログラムされない状態のままにできる。

【0297】PT+40およびPT+44にはそれぞれ第11および第12データワードがある。次元の定められた packets 転送では、これら32ビットフィールドでは、ソースまたは宛て先の第3次元のピッチを特定する。次のパッチの開始アドレスを得るのに、ソースまたは宛て先パッチの開始アドレスにこのピッチの値が加算される。対応するCカウントフィールド内の値が0であれば、このフィールドはプログラムされない状態のままにできる。

【0298】デルタまたはオフセットガイド転送において、第11および第12データワードは、ガイドテーブルアドレスである。これらフィールドにはそれぞれのガイドテーブル内の第1エントリーをポイントする整列されたアドレスがロードされる。これらデータワードの値は、ガイドテーブルからエントリーが取り出されるたびに固定パッチ転送に対しては4だけ、または可変パッチ転送に対しては8だけ自動インクリメントされる。これらガイドテーブル開始アドレスは、固定パッチ packets 転送に対しては32ビットワードアドレスに対して整列し、可変パッチ packets 転送に対しては64ビットダブルワードアドレスに整列しなければならない。ガイドテーブルはオンチップメモリ内になければならず、いずれかの条件が真でなければ、イリーガルなアドレスエラーインタラプトが生じる。

【0299】第9および第11データワードは、値で満たされた packets 転送において特別な用途がある。これら2つの32ビットフィールドは64ビットフィルパターンを構成する。宛て先のダブルワードに書き込まれたバイトはフィル値のダブルワード内の対応するバイトから取り出される。従ってフィルパターンが64ビット未満であれば、これら2つのデータワード内でデータ値を繰り返さなければならない。値でフィルする packets 転送に対し、このフィル値はソースに対してのみ可能であ

るが、他方、宛て先は次元の定められた、固定または可変パッチのデルタまたはオフセットガイドにできる。

【0300】 packets 転送がソーストランスペアレンシーを利用する場合、第13および第14データワードはトランスペアレンシー値をホールドする。この64ビットのダブルワードはPT+48で開始する。この64ビットフィールドはトランスペアレンシーを選択した場合、比較すべき値を表示するのに使用される。 packets 転送オプションフィールドの packets アクセスモードフィールドは、このフィールド内に含まれる値のトランスペアレンシーサイズまたは数を表示する。宛て先ダブルワードにすぐに書き込まれるバイトは、トランスペアレンシー値内の対応するバイトと比較され、一致が見つかればバイトストローブが非アクティブにドライブされる。ピクセルサイズが64ビット未満であれば、データ値を再現しなければならない。このフィールドはブロックの書き込みを除くすべての非トランスペアレンシー転送に対しては用いられない。ソーストランスペアレンシー値は、エンディアンモードにかかわらず、メモリに書き込まれるのとまさに同じように、転送コントローラ80によって使用されるので、ワードスワップは生じない。これは図25に示されている。 packets 転送パラメータテーブル開始アドレスに続く48バイトをアドレス指定するダブルワード書き込みとして常にこの値を書き込むことにより、混乱を避けることができる。

【0301】 packets 転送はブロック書き込みモードを使用する場合、第3および第4データワードはカラーレジスタのデータをホールドする。64ビットのカラーレジスタフィールドはブロック書き込みサイクルの準備の際にビデオランダムアクセスメモリ (VRAM) をロードするのに使用される値を含む。この値は、シミュレートされたブロック書き込みを実行する際にも使用される。転送コントローラ80はエンディアンモードにかかわらず、トランスペアレンシー値に関してこれまで述べたのとまったく同じ態様で書き込まれるカラーレジスタ値を使用する。これは図29に示されている。以下、このフィールドの使用について詳細に説明する。

【0302】 packets 転送パラメータの最終ダブルワードを形成するPT+56で開始する第15および第16データワードは、すべての転送モードに対してそのときは使用されず、プログラムされない状態のままにできる。これらデータワードはマルチプロセッサ集積回路100を未来バージョンで使用できる。

【0303】図25~29は、 packets 転送パラメータテーブルの少数の可能なフォーマットを示す。図25は、ソーストランスペアレンシーを有するソースおよび宛て先用の次元の定められた packets 転送用のパラメータのテーブルの一例を示し、PT+48にトランスペアレンシーワード0が記憶され、PT+52にトランスペアレンシーワード1が記憶される。図26は、値でフィ

ルする宛て先用の次元の定められたパケット転送用パラメータテーブルの一例を示す。PT+40にフィル値の最大のビットが記憶される。PT+32にフィル値の最小位のビットが記憶される。図27は、次元の定められたソースおよび固定パッチガイド宛て先パケット転送用のパラメータテーブルの一例を示す。PT+12におけるデータワードは、ベースアドレスを記憶し、PT+28におけるデータワードはガイドテーブル内のエン트리数を記憶し、PT+44におけるデータワードはガイドテーブルの開始アドレスを記憶する。図28は、次元の定められたソースおよび可変パッチガイド宛て先パケットのためのパラメータテーブルの一例を示す。通常PT+22に記憶される宛て先AおよびBカウンタはプログラムされないが、その理由は、これらカウンタが宛て先ガイドテーブルの一部であるからである。図29は、次元の定められたブロック書き込みパケット転送用の一例を示す。PT+48で開始するダブルワードは、カラーレジスタワード0および1を含むことに留意されたい。

【0304】パケット転送オプションフィールドはソースおよび宛て先転送のためにどんなフォームの転送を用いるかを選択し、現在のパケットがリンクされたリストを終了させるかどうかを決定する。更に特殊転送モード、パケット転送終了時の付加的地址計算の実行、次元アドレス計算の方向変化またはソース転送と宛て先*

*転送の反転のような多数の付加的特徴の選択を可能にする。図30にパケット転送オプションフィールドのフォーマットが示されている。オプションフィールドにすべて0をロードすることにより、特殊アドレスモードを用いないで、ソースおよび宛て先で次元の定められた転送を利用するデフォルトパケット転送を行うことになる。

【0305】パケット転送オプションフィールドのビット1~0は、宛て先更新モードフィールドを形成する。これら2つのビットは、元のパケット転送パラメータテーブル内の宛て先開始アドレスが、パケット転送が完了した場合にどのように更新されるかを表示する。これらビットが0でなければ、パケット転送完了後に余分なアドレス計算を行う。パケット転送パラメータテーブルに指定された元の宛て先開始アドレス上にこの値が書き込まれる。これによりパケット転送を更に送ることが可能となり、オフ状態から続けることが可能となる。これは特に2つのメモリエリア間でピンポン操作するのに有効である。

【0306】表15に、宛て先更新モードがリストアップされている。下記のような宛て先反転アドレス指定ビットの反転宛て先Cまたは反転宛て先Bの一方がセットされると、加算よりも減算を行う。

【0307】

【表15】

ビット		宛て先更新オペレーション
1	0	
0	0	更新せず
0	1	宛て先転送内の最終ラインの開始アドレスにBピッチを加算（これより減算）し、結果を宛て先に書き込む。
1	0	宛て先転送内の最終パッチの開始アドレスにCピッチを加算（これより減算）し、結果を宛て先に書き込む。
1	1	宛て先転送内の最終パッチの開始アドレスにCピッチを加算（これより減算）し、その結果をPTパラメータテーブル内の宛て先および開始アドレスに書き込み、次にPTオプションフィールド内の反転宛て先Cアドレス指定ビットをトリグリングする。

表15

【0308】宛て先更新モードは、使用中の最大次元の追加ステップを実行するのに使用できるが、他の有効なオペレーションも実行できる。転送は2次元だけに行うことができたが、Cピッチを加えるよう、モード10も使用できる。これにより最終パケットに対し次の2次元パケットを位置させることができる。2つの1次元または2次元パッチ間で、例えばオンチップデータメモリ上でピンポン動作するパケット転送を再送信するのにモード11が特に有効である。完了時にパケット転送パラメータを更新するたびに、第3次元のアドレス指定方向が反転されるので、この方向はメモリエリア間で交互に変わる。

【0309】この機能は、主に次元の定められた転送と

共に使用することを意図するものである。この機能は、ガイド転送で指定できるが、注意が必要である。ガイド転送に対しては、宛て先Cピッチは宛て先ガイドテーブルポイントと置換されていることを思い出していただきたい。

【0310】パケット転送オプションフィールドのビット6~4は、宛て先転送モードフィールドを形成する。これら3つのビットは宛て先アドレス指定のためにどのフォームの転送を使用すべきかを表示する。ソース指定モードルックアップテーブルおよび値によるフィル操作は宛て先アドレス指定のために定義されていない（保留されている）。

【0311】

【表16】

ビット 6 5 4			宛て先転送モード
0	0	0	次元の定められた転送
0	0	1	保留
0	1	0	保留
0	1	1	保留
1	0	0	可変パッチデルタガイド転送
1	0	1	可変パッチオフセットガイド転送
1	1	0	固定パッチデルタガイド転送
1	1	1	固定パッチオフセットガイド転送

表16

【0312】パケット転送オプションフィールドのビット9～8は、ソース更新モードフィールドを形成する。*反転ソースBがセットされる場合、加算よりも減算を実行する。
表17に、このソース更新モードが示されている。適当 【0313】
なソース反転アドレス指定ビット、反転ソースCまたは* 【表17】

ビット 9 8		ソース更新オペレーション
0	0	更新せず
0	1	ソース転送内の最終ラインの開始アドレスにBビットを加算（これより減算）し、結果をソースに書き込む。
1	0	ソース転送内の最終パッチの開始アドレスにCビットを加算（これより減算）し、結果をソースに書き込む。
1	1	ソース転送内の最終パッチの開始アドレスにCビットを加算（これより減算）し、その結果をPTパラメータテーブル内のソースおよび開始アドレスに書き込み、次にPTオプションフィールド内の反転ソースCAアドレス指定ビットをトリグリングする。

表17

【0314】これら2つのビットはパケット転送が完了した際に元のパケット転送パラメータ内のソース開始アドレスをどの値で更新すべきかを表示する。これらビットが0でなければパケット転送完了後に追加的ソースアドレス計算を実行する。次にパケット転送パラメータに指定された元のソース開始アドレス上にこの値を書き込む。これによりパケット転送を再び送ることができ、先の時間に残っていた場所より続けることができる。このような転送は、2つのメモリエリア間でピンポン操作するのに特に有効である。これらモードは先に宛て先アドレス※

※レスの発生で述べたように、操作アドレスの発生でも同じような用途がある。

【0315】パケット転送オプションフィールドのビット14～12は、ソース転送モードフィールドを形成する。これら3つのビットは、ソースアドレス指定のためどのフォームの転送を使用すべきかを表示する。表18にこれらのコーディングを示す。

【0316】

【表18】

ビット 14 13 12			ソース転送モード
0	0	0	次元の定められた転送
0	0	1	値でフィルする転送
0	1	0	保留
0	1	1	固定パッチオフセットガイドLUT
1	0	0	可変パッチデルタガイド転送
1	0	1	可変パッチオフセットガイド転送
1	1	0	固定パッチデルタガイド転送
1	1	1	固定パッチオフセットガイド転送

表18

【0317】パケット転送オプションフィールドのビット18～16は、パケット転送アクセスモードフィールドを形成する。これら3つのビットは表19に示されるような特別なアクセスモードをエンコードするのに用い

られる。これらモードは宛て先にソースデータを書き込む方法を変えるものであり、オンチップメモリの宛て先に対し、モード000の通常の転送しか許可されない。

【0318】

【表19】

ビット			パケット転送アクセスモード
18	17	16	
0	0	0	通常ページモード
0	0	1	周辺デバイス転送
0	1	0	ブロック書き込み
0	1	1	シリアルレジスタ転送
1	0	0	8ビットソースストランスペアレンシー
1	0	1	16ビットソースストランスペアレンシー
1	1	0	32ビットソースストランスペアレンシー
1	1	1	64ビットソースストランスペアレンシー

表19

【0319】000のコーディングは、外部メモリに対する通常のアクセスモードまたはページモードアクセスを可能にする。ソースまたは宛て先上では特別なアドレス指定モードは使用されない。変更をすることなく、ソースから宛て先にデータが転送される。

【0320】001のコーディングは、周辺デバイス転送モードを可能にする。周辺デバイスモードはメモリコントローラとして転送コントローラ80を使用するマルチプロセッサ集積回路100の外部のメモリを、他のデバイスが読み出ししたり書き込んだりするのを可能にする。デバイスのメモリの読み出しはソースをプログラムすることによって達成され、デバイスのメモリへの書き込みは宛て先をプログラムすることによって達成される。いずれの場合にせよ、転送コントローラ80は通常は転送パラメータに従って発生されたアドレスによりメモリアドレスおよび制御ラインをドライブするが、周辺デバイスがデータを読み出ししたりドライブしたりできるように、データバスをハイインピーダンスモードとする。周辺デバイスモードは値でフィルする転送を除き、任意の形態のソースまたは宛て先転送と共に使用できる。

【0321】010のコーディングはパケット転送がVRAMブロック書き込みを使用できるようにする。これにより、転送コントローラ80はVRAMカラーレジスタにロードし、VRAMブロック書き込みモードを使って外部メモリへの宛て先書き込みを実行する。このモードでは、VRAMカラーレジスタデータがVRAM内のどの位置に書き込まれるかを指定する、ブロック書き込みアドレスマスクビットをソースデータが表示する。これらビットは通常のアドレス指定を用いてソースからフェッチされ、その後、ブロック書き込みモードを用いて宛て先VRAMに書き込まれる。VRAMカラーレジスタにロードされる値は、パケット転送パラメータのカラーレジスタ値として指定される。

【0322】ブロック書き込みオペレーションはオフチップの宛て先だけにサポートされている。オンチップ宛て先アドレスへのブロック書き込みの試みは、エラー条件によりパケット転送を一時中断させる。

【0323】011のコーディングはシリアルレジスタ

転送モードを可能にする。一般に、VRAMのバルク初期化を行うために、シリアルレジスタ転送モードが使用される。VRAMの行をVRAMシリアルシフトレジスタにコピーするのに、ソースアドレスが使用される。次にシフトレジスタを多数のVRAMメモリの行にコピーするのに、宛て先アドレスが使用される。通常のオペレーションはソースへのカウントに1バイトをロードし、宛て先に1のAカウントおよびn-1（ここでnは書き込みを行う行の数）のBカウントをロードすることである。このモードでは、データバスまたはクロスバー50上でのデータ転送は行われず、VRAMによりすべてのデータ転送が行われることに留意されたい。すべてのソースアクセスおよび宛て先アクセスは非ページモードとなる。シリアルレジスタ転送オペレーションは、オフチップのソースおよび宛て先だけにサポートされている。

【0324】1XXフォームのコーディングはトランスペアレンシーを可能にする。通常、ソースおよび宛て先の次元の定められた転送またはガイド転送が実行される。しかしながら宛て先データを書き込む前に、このデータはパケット転送パラメータ内に示されたトランスペアレンシー値を比較される。このトランスペアレンシーの比較は、トランスペアレンシーおよびバイト書き込み回路353で行われる。パケットアクセスモードの2つの最小位のビットはトランスペアレンシーデータのサイズを表示する。従って、1回の64ビットの比較、2回の32ビットの比較、4回の16ビットの比較または8回の8ビットの比較がなされる。比較のいずれかが真であれば、トランスペアレンシーおよびバイト書き込み回路353は、宛て先バイトが書き込まれないように対応するバイトストローブをディスエーブルする。オンチップソースまたは宛て先アドレスによるトランスペアレンシーの試みにより、パケット転送はエラー条件で一時中断される。

【0325】パケット転送オプションフィールドのビット19は、交換用ソースおよび宛て先パラメータビットである。ビット19のセットにより、ソースおよび宛て先パラメータをマチュアルでスワップすることなく、パケット転送方向を反転できる。これは元の位置にデータを戻す際に有効である。交換用ビットがセットされる

20

30

40

50

101

と、転送コントローラ80はバケット転送パラメータをロードする際に、すべてのソースおよび宛て先値の開始アドレス、ピッチ、カウント、ガイドテーブルポインタおよびフィル値のすべてをスワップする。表20は、このビットがセットされる場合の32ビットのスワップを*

バイト アドレス		バイト アドレス
ソース開始アドレス	PT+08 <=>	PT+12 宛先開始アドレス
ソースA/Bカウンタ	PT+16 <=>	PT+20 宛先A/Bカウンタ
ソースピッチ	PT+24 <=>	PT+28 宛先ピッチ
ソースBピッチ	PT+32 <=>	PT+36 宛先Bピッチ
ソースCピッチ	PT+40 <=>	PT+44 宛先Cピッチ

102

* 示す。PTはバケット転送パラメータテーブルの次のエントリーアドレスのアドレスを表示する。

【0326】

【表20】

表20

【0327】PTにおける次のエントリーアドレスデータおよびPT+04におけるバケット転送オプションフィールドは、これらの値がソースまたは宛て先に関連していないので、それらの先の位置に止まることに留意されたい。更にトランスペアレンシーデータまたはカラーレジスタデータを記憶するPT+48およびPT+52は、スワップされないことにも留意されたい。これにより、64ビットのトランスペアレンシーまたはカラーレ※20

※ジスタパラメータはその値を維持できる。転送コントローラ80はソースおよび宛て先に関連したパラメータワードのスワップのほかに、バケット転送オプションフィールド内でソースおよび宛て先に関連したビットもスワップする。これについては表21に示されている。

【0328】

【表21】

ビット 番号		ビット 番号
宛先更新モード	0 <=>	8
宛先更新モード	1 <=>	9
宛先更新モード	4 <=>	12
宛先更新モード	5 <=>	13
宛先更新モード	6 <=>	14
宛先更新モード	24 <=>	21
宛先更新モード	25 <=>	22

表21

【0329】スワップのいずれかの結果、機能がサポートされなくなると、バケット転送はエラー条件により一時中断する。バケット転送パラメータがロードされる時はいつも、ソースパラメータと宛て先パラメータとの交換が実行される。バケット転送が一時中断されている場合、そのときのパラメータはリクエスト中のプロセッサのパラメータメモリにセーブされる前に、元の位置へスワップし戻される。一時中断されたバケット転送がレストアされる場合、パラメータは転送コントローラ80によりロードされる際に再びスワップされる。

【0330】バケット転送オプションフィールド内で更新モードのうちの一つが指定される場合、バケット転送の完了時に通常のオペレーションが行われる。例えばソース更新オペレーションが選択される場合、元のバケット転送パラメータ内ソース開始アドレスは、バケット転送中に宛て先開始アドレスとして実際に使用されても更新されることになる。同様に、更新モードとしてトグル反転ソースCアドレス指定ビットを指定すると、元のバケット転送オプションフィールドのビット22がトグルされ、これにより実際にバケット転送が再送信され

た場合、宛て先Cのアドレス指定が反転される。

【0331】上記例が示すようにXビットを使用する際には注意が必要である。例えば次元の定められた宛て先バケット転送に対する値で満たされたソースのために指定する場合、値によるフィルオペレーションは宛て先オペレーションとして指定できないので、バケット転送はエラーにより一時中断される。エラーが発生しない場合でも、宛て先開始アドレスおよびカウントには、一般にプログラムされないソース値がロードされる。同様に、フィル値のワードには宛て先ピッチがロードされる。この結果、極めて無意味なバケット転送となる。

【0332】バケット転送オプションのビット21は反転ソースBアドレス指定ビットである。このビットを1にセットすると、ソースの第2の次元が後方にアドレス指定される。加算よりもむしろ先のライン開始アドレスからBピッチが減算される。このビットが0であれば、通常はバケット転送ソースアドレス指定が行われる。

【0333】バケット転送オプションフィールドのビット22は、反転ソースCアドレス指定ビットである。このビットを1にセットすると、ソースの第3次元が後方

にアドレス指定される。加算よりもむしろ先のパッチ開始アドレスからCピッチが減算される。このビットが0であれば、通常はパケット転送ソースアドレス指定が行われる。このビットはガイド転送に対しては意味がなく、ガイドテーブルを介してCのアドレス指定を特定する。

【0334】パケットテーブルオプションフィールドのビット23は、反転ソースAアドレス指定ビットである。このビットを1にセットすると、ソースおよび宛て先の第1次元は後方にアドレス指定される。ダブルワードアドレスはインクリメントされる代わりにデクリメントされる。このことは、Bピッチが加算（または減算）される値は、第1の次元における最も大きなアドレスであることを意味している。ダブルワード内でのバイトによるアドレス指定は反転されず単なるダブルワードのアドレス指定であることに留意されたい。

【0335】パケットテーブルオプションフィールドのビット24は、反転宛て先Bの宛て先指定ビットである。このビットを1にセットすると、宛て先の第2の次元が後方にアドレス指定される。Bピッチは加算されるよりもむしろ先のライン開始アドレスから減算される。このビットが0であれば、通常はパケット転送ソースアドレス指定が行われる。

【0336】ビット25は反転宛て先Cのアドレス指定ビットである。このビットを1にセットすると、宛て先の第3の次元が後方にアドレス指定される。Cピッチは加算されるよりもむしろ先のパッチ開始アドレスから減算される。このビットが0であれば、通常はパケット転送ソースアドレス指定が行われる。このビットはガイド転送に対しては意味がなく、ガイドテーブルを介してC

アドレス指定を特定する。
【0337】パケットテーブルオプションフィールドのビット28は、ビットを終了した際のインタラプトである。このビットを1にセットすると、リンクされたリスト上のこのエントリーを終了した後すぐに、パケット*

* 転送を開始しているプロセッサに通常のインタラプトが送られる。リンクされたリストは更にエントリーを含むことができる。これによりリンクされたリスト内の特定のポイントに達したときに、リクエスト中のプロセッサにフラグを立てることができる。このビットが0であれば、リンクされたリスト内の対応するパケットエントリーが終了した際には、プロセッサにはインタラプトは送られない。従って、このビットセットを有するエントリーが生じて完了するか、またはリンクされたリストの終わりにあるパケット転送が完了するかのいずれかまでに、インタラプトが発生されることはない。しかしながらエラーがある時間に生じた場合、転送コントローラ80はリクエスト中のプロセッサに即座にエラーインタラプトを送る。

【0338】パケット転送オプションフィールドのビット30～29は、パケット転送ステータスフィールドを形成する。パケット転送ステータスフィールドはパケット転送リクエストのステータスを表している。プロセッサがリクエストを送る際には、これらビットは常に00とセットされていなければならない。リンクされたリスト内のパケット転送が一時中断状態の場合、転送コントローラ80は、これが中断中のパケットパラメータエリアにセーブしているパケット転送オプションフィールド内の適当なパケット転送ステータスビットをセットする。中断されたパケット転送がフレッシュなパケット転送よりも多くのパケットを含んでおり、従って異なる態様でセーブされ、レストアされるので、このような操作が必要である。転送コントローラ80がパケット転送パラメータをロード中に、これらビットのいずれかが1とみなされると、転送コントローラはパケット転送を一時中断し、よって余分なパラメータのすべてをレストアする。このフィールドは、表22にリストされているようにコード化される。

【0339】

【表22】

ビット 30 29		パケット転送ステータス
0	0	一時中断されていない
0	1	一時中断であるがフォールトでない
1	0	一時中断、ソース上でフォールト
1	1	一時中断、宛て先上でフォールト

表22

【0340】次の条件のうちの1つが生じたためにパケット転送が一時中断されると、転送コントローラ80はこれらビットに01を書き込む。すなわち転送コントローラ80により、より高い優先度のパケット転送リクエストが受信される場合、パケット転送が多くタイムアウトしてしまった場合、パケット転送をリクエストしたプロセッサがパケット転送の一時中断を求めた場合、また

はエラー条件が生じた場合、ビット30は実際にフォールトが生じたことによりパケット転送を一時中断したことを表示する。ビット29は、フォールトがソースにあるのか宛て先にあるのかを表示する。フォールト条件を解決するため、この情報はマスタプロセッサ60が必要とする。このことは、フォールトの生じたパケット転送のパラメータを再ロードする際の転送コントローラ80

には関係がない。その理由は、一時中断プロセスとレストアプロセスとはすべてのタイプの一時中断パケット転送に対して同じであるからである。

【0341】交換用ソースおよび宛て先パラメータビット（ビット19）が中断されたパケット転送オプションフィールド内で1であれば、フォールトの生じたパケット転送に対するビット29の意味は反転する。従って10が宛て先上でのフォールトを示し、11がソース上でのフォールトを示す。従ってマスタプロセッサ60のソフトウェアはフォールトの生じたアドレスを決定する際に、ビット29と19の双方を検査しなければならない。

【0342】転送コントローラ80はパケット転送中にフォールトが生じた場合に数種の措置をとる。転送コントローラ80は、ビット29と30を適当にセットし、この適当なビットをFLTSTSレジスタ内にセットする。転送コントローラ80がマスタプロセッサ60にフォールトインタラプトを発生する。デジタル画像/グラフィックプロセッサ71、72、73または74のうちの 하나가パケット転送を発生する場合、このプロセッサはフォールトが生じていることについて認識していない。マスタプロセッサ60は、このフォールト条件をクリアし、FLTSTSレジスタ内のフォールトフラグをクリアしなければならない。FLTSTSレジスタ内の対応するフォールトフラグをクリアする際に転送コントローラ80はパケット転送を自動的に再送信する。

【0343】パケット転送オプションフィールドのビット31はストップビットである。このビットはリンクされたリストの終了部をマークするのに使用されている。このビットが1であるパケット転送に会うと、パケット転送は完了され、リンクされたリストが終了される。終了前にリクエスト中のプロセッサのパラメータメモリ内のリンクされたリストの開始アドレス位置に、次のエントリーアドレスフィールドがコピーされる。従ってリンクされたリストが再イネーブル化されると、リンクされたリスト内の次のエントリーで実行が開始する。これにより、ピンポンオペレーションのような繰り返しオペレーションに特に有効な円形状のリンクされたリストが形成できる。ストップビットは所望の位置でリンクされたリストをブレイクするのに使用できる。

【0344】転送コントローラ80がパラメータテーブルからのパケット転送パラメータを一旦読み出すと、データの転送を開始する準備が完了する。これを行うため、転送コントローラ80はクロスバアクセスまたは外部メモリアccessのいずれかまたは双方を発生しなければならない。ソースから宛て先へのデータの基本的フローには4つの可能性がある。すなわちオンチップからオンチップへ、オンチップからオフチップへ、オフチップからオンチップへ、更にオフチップからオフチップへのフローがある。最初の3つは通常取り扱われている

が、最後のケースは特別なケースである。

【0345】ソースマシン320は、正常なパケットデータ転送フロー中に、転送パラメータに基づきソースアドレスを発生し、これを用いて適当なオンチップまたはオフチップメモリからデータをフェッチする。クロスバまたは外部メモリバスからデータが受け取られると、必要なバイトは抽出され、ソース整列ロジック332によって整列され、次にパケット適当なFIFOバッファ312に入れられる。これと同時に宛て先マシン340は宛て先メモリのためのアドレスも発生する。パケット転送FIFOバッファ311が一旦次の宛て先メモリアccessに必要なバイト数を含むと、宛て先マシン340は必要なクロスバまたは外部メモリアccessを発生する。

【0346】パケット転送FIFOバッファ311は、データフローを制御し、ソースマシン320と宛て先マシン340との同期を維持するように働く。あるときに、パケット転送FIFOバッファ311が次の宛て先accessのための十分なソースバイトを含んでいない場合、データが利用可能となるまで宛て先マシン340は停止する。これと同じように、パケット転送FIFOバッファ311がフル状態となれば、宛て先マシン340が次のソースaccessを完了させるのに十分なバイトを引き出すまで停止する。これにより、ソース転送が宛て先転送をオーバーランさせることが防止される。特殊なパケット転送accessモードに対しては、このデータフローは多少変えることができる。

【0347】クロスバ50と外部メモリアccessインターフェースは独立しているため、ソース転送は外部バス上で行われ、クロスバ上で宛て先転送を行うことができ、また、パラレル状態で外部バス上で宛て先転送を行い、クロスバ上でソース転送を行うこともできる。オンチップからオンチップへの転送に対してはソースと宛て先とは必要に応じてクロスバインターフェースを共用し、サイクルをインターリーブすることになる。

【0348】キャッシュサービスリクエスト、ダイレクト外部accessリクエスト、フレームコントローラ90のリクエスト、緊急リフレッシュおよびホストリクエストは、パケット転送を一時中断させることはない。これらリクエストがパケット転送よりも優先度が高い場合にのみ、ソースマシン320および宛て先マシン340の一方または双方を停止できる。フレームコントローラ90および緊急リフレッシュサイクルは、外部メモリアccessインターフェースだけを使用する。従ってパケット転送クロスバaccessは続いて行うことができる。ソースマシン320または宛て先マシン340のいずれかが、外部メモリアccessインターフェースを使用している場合、パケット転送FIFOバッファ311は、最終的にフル状態または空状態となる。従ってクロスバを用いるコントローラは外部メモリアccessインターフェースが再び利用できるまで

停止していなければならない。オンチップからオンチップへの転送である場合、ソースマシン320および宛て先マシン340の双方は、障害を受けない状態を続けることができる。キャッシュおよびダイレクト外部アクセスサービスは、クロスバーと外部インターフェースの双方を使用するので、これらは一般に、必要とするサイクル数の間でパケット転送を停止する。

【0349】オンチップソースからオフチップの宛て先へのパケット転送は、他の3つのパケット転送の例と異なっており取り扱われる。DRAMまたはVRAM上でページモードの利点を活用するため、転送コントローラ80はオフチップソースからオンチップへの列アクセスのページモードバーストを実行し、次にオンチップからオフチップの宛て先への別のページモードバーストを実行する。これを行うにはオンチップバッファを使用しなければならない。各プロセッサは図15に示すように、この目的のために保留された対応するパラメータメモリ内の128バイトのエリアを有する。図15に示すように、マスタプロセッサ60によってリクエストされるオフチップからオフチップへのパケット転送は、アドレスの16進数01010100から16進数0101017Fを使用する。デジタル画像/グラフィックプロセッサ71、72、73、74のうちの一つによってリクエストされるオフチップからオフチップへのパケット転送が、アドレスの16進数0100#100から16進数0100#17F（ここで#は表3にリストされているようなデジタル画像/グラフィックプロセッサの番号に対応する）を利用する。このパラメータメモリバッファの内外へのデータの転送は、転送コントローラ80のハードウェアにより処理され、ユーザーに対しトランスペアレントである。

【0350】パケット転送パラメータの組み合わせは、完了するのに極めて長い時間を必要とする、極めて大きな転送の指定を可能とする。更により高い優先度のパケット転送リクエストが、パケット転送をインタラプトし続け、パラメータのロードに長い時間がかかるの防止する。このような状況のいずれかの発生を防止するため、転送コントローラ80は図31に示されている2つの24ビットレジスタのパケット転送最小レジスタPTMIN511とパケット転送最大レジスタPTMAX512を含むタイマーを有する。これらレジスタはパケット転送の最小長さと最大長さを指定する。パケット転送カウンタPTCOUNT513は、パケット転送を実行するクロックサイクル数を表示する。

【0351】パケット転送最小レジスタPTMIN511は、より高い優先度のパケット転送リクエストによって中断できる前に、パケット転送が実行しなければならないクロックサイクルの最小数を示す。パラメータがロードされた後にパケット転送が開始すると、タイマーシーケンサ515はパケット転送最小レジスタPTMIN

511に記憶されていた値を、パケット転送カウンタPTCOUNT513にロードする。パケット転送カウンタPTCOUNT513は、パケット転送が転送コントローラ80によってアクティブにサービスされるクロックサイクルごとに、1だけデクリメントする。パケット転送カウンタPTCOUNT513は、パケット転送中に生じ得るキャッシュサービスサイクル、フレームコントローラ90サイクル、ホストサイクルまたはリフレッシュサイクル中にはデクリメントしないが、アクティブなパケット転送サービス中に生じる再試行または待機ステート中にデクリメントする。しかしながら、クロスバーのパケット転送アクセスがまだ行われている場合、外部バス上でのフレームコントローラ90、ホストまたはリフレッシュ活動中にデクリメントされる。パケット転送はPTCOUNTが0に達したことを0検出器514がタイマーシーケンサ515に信号を送るまで、優先度の高いパケット転送リクエストによってインタラプトすることはできない。またパケット転送はエラーまたはフォールト条件によって一時中断することもできる。パケット転送最小レジスタPTMIN511には、リセット時に16進数10000（65,536サイクル）がロードされる。

【0352】パケット転送最小レジスタPTMIN511の重要な用途は、すでに中断したパケット転送のパラメータを別のメモリエリアに転送するための停止不能なパケット転送を行うことである。別の一時中断によりデータを転送するパラメータメモリエリアにオーバーライトされるので、パケット転送最小レジスタPTMIN511はパケット転送を完了できるように保証するために使用される。かかる転送はフォールト状態にならないことが重要である。その理由は、この状態になるとパラメータメモリエリアもオーバーライトされるからである。このような特徴により、最小時間前により高い優先度のメモリアccessがパケット転送をアボートすることが防止され、従って最小の数のデータ転送が行われる。

【0353】パケット転送最大レジスタPTMAX512は、1つのパケット転送によるデータの転送の独占を防止するのに使用される。PTMIN時間が経過し、パケット転送カウンタPTCOUNT513が0検出器514に検出されるように、0にデクリメントされると、タイマーシーケンサ515はパケット転送最大レジスタPTMAX512内に記憶されていた値をパケット転送カウンタPTCOUNT513にロードする。これにより、タイムアウトするまでに転送を進めることができる残りの時間が決まる。従ってインタラプトされないパケット転送のための最大期間は、PTMIN+PTMAXクロックサイクルとなる。パケット転送カウンタPTCOUNT513は、リフレッシュのような非パケット転送サイクルを除く、パケット転送が連続的にアクティブとなるサイクルごとにデクリメントされる。パケット転

送完了前にパケット転送カウンタPTCOUNT513内の値が0に達したことを、0検出器514が検出すると、パケット転送はタイムアウトしたものと見なされる。タイマーシーケンサ515は、パケット転送を一時中断し、転送コントローラ80はラウンドロビン状に同じ優先度の次のリクエストに移る。同じ優先度の他のリクエストがペンディング中となっていなければ、このような同じ優先度のリクエストまたは優先度のより高いリクエストが生じるまで、または転送が完了するまで、0のPTCOUNTと共に転送を続行できる。パケット転送最小レジスタPTMIN511内に指定されたサイクル数が経過した後に、優先度のより高いパケット転送リクエストが受信されると、PTMAXに達したか否かに係わらず、アクティブなパケット転送が一時中断される。パケット転送最大レジスタPTMAX512にはリセットで16進数10000(65,536サイクル)がロードされる。

【0354】パケット転送が一時中断状態となると、リンクされたリスト全体も一時中断される。ラウンドロビントークンはリンクされたリスト内の次のパケット転送でなくて、ペンディング中のリクエストと共に次のプロセッサに進む。パケット転送がタイムアウトすると、一時中断された転送のステートはリクエスト中のプロセッサのパラメータメモリにセーブされる。ラウンドロビンの優先度がそのプロセッサに戻されると、転送コントローラ80は連続のためのこのリクエストを自動的に再送信する。一時中断されたパケット転送が再開されるときはいつも、フル状態のPTMINおよびPTMAX値が有効となる。

【0355】好ましい実施例では、パケット転送最小レジスタPTMIN511およびパケット転送最大レジスタPTMAX512は、24ビットしか含まない。従ってタイムアウトすることなく、パケット転送サービスが続くことのできる最大時間は50MHzの目標動作周波数で約0.67秒である。

【0356】パケット転送用のリンクされたリストは、次のように転送コントローラ80によって管理される。緊急優先度を除くパケット転送中にリフレッシュ、フレームコントローラ90、ホストインターフェースまたはキャッシュサービスリクエストが受信される場合、パケット転送パラメータのステートは転送コントローラ80の内部レジスタに保持され、必要であればソースおよび/または宛て先転送が停止される。優先度のより高いリクエストのサービスが完了すると、パケット転送が開始される。

【0357】優先度のより高いパケット転送リクエスト、タイムアウト、リクエスト中のプロセッサからの一時中断リクエスト、フォールトまたはエラーによってパケット転送がインタラプトされると、パケット転送が一時中断される。リクエスト中のプロセッサのパラメータ

メモリ内のリンクされたリストの開始アドレスが、セーブされたパケット転送パラメータをポイントするように変更される。このインタラプトが優先度のより高いパケット転送によるものである場合、ラウンドロビントークンはインタラプトされたパケット転送と共に留まるので、優先度のより低いリクエストが再開されると、そのサービスが再開される。パケット転送パラメータのローディング中に優先度のより高いリクエストが生じると、ローディングが停止される。一時中断は行われない。優先度のより高いパケット転送が完了すると、元のパラメータリストからパケット転送パラメータがロードされる。パケット転送がタイムアウト、フォールト、エラーまたは一時中断リクエストにより一時中断されている場合、優先度決定チェーンの終了部にインタラプトされたパケット転送を送るように、ラウンドロビントークンが進められる。

【0358】パケット転送が完了し、パケット転送オプションフィールドのインタラプトビットが1となると、転送コントローラ80はリクエスト中のプロセッサにパケットの終了部のインタラプト信号を発生する。リンクされたリスト内に1のパケット転送オプションフィールドの停止ビットを有する最終パケット転送が完了すると、転送コントローラ80がリクエスト中のプロセッサにパケットの終了部インタラプト信号を発生する。パケット転送が完了すると、パケット転送のパケット転送オプションフィールドストップビットがセットされていても、リクエスト中のプロセッサのパラメータメモリ内のリンクされたリストの開始アドレス位置に、パケット転送パラメータからの次のアドレスフィールドが書き込まれる。

【0359】パケット転送パラメータがロードされると、転送コントローラ80はパケット転送オプションフィールドのパケット転送ステータスビットをチェックする。パケット転送が一時中断されたことをこれらビットが表示する場合、一時中断された転送を付加的ステート情報がロードされる。これらについては後に更に説明する。

【0360】パケット転送がエラー状態を経験すると、転送コントローラ80は即座にデータの転送を停止し、一時中断を実行する。転送コントローラ80はリクエスト中のプロセッサのPTERRORフラグをセットする。転送コントローラ80はリンクされたリストが終了したことを表示するため、リクエスト中のプロセッサのCOMNレジスタ120のQビットもセットする。リクエスト中のプロセッサは、セーブされたパラメータからのエラーの原因を決定するようにプログラムすることができる。

【0361】パケット転送中に多数の条件のいずれかがエラーを生じさせる。試みられたVRAMアクセスモード、例えばオンチップメモリ内のソースまたは宛て先に

10

20

30

40

50

よるブロック書き込み宛て先転送またはシリアルレジスタ転送がエラーを生じさせる。トランスベアレンシーを利用して試みられるオンチップ宛て先アクセスもエラーを生じさせる。値でフィルされた場合を除くソース転送、シフトレジスタ転送または周辺デバイス転送の長さよりも、パケット転送リクエスト宛て先転送のバイトの総数の長さが長くなると、エラーが生じる。ルックアップテーブルの宛て先オペレーションを試みる際にエラーが生じる。このようなエラーは、交換用ソースおよび宛て先パラメータのオペレーションから生じる。リンクされたリストの開始アドレスまたはリンクされたリストの次のエントリがアドレスのオフチップをポイントする際にエラーが生じる。パケット転送パラメータテーブルおよびリンクされたリストの開始アドレスが64バイトの境界に整列していないときにも、エラーが生じる。ソースまたは宛て先ガイドテーブルポインタがオフチップメモリをポイントしたり、これらが正しく整列されていないときに試みられるパケットリクエストの際にエラーが生じる。実際のエラー状態を表示するようにステータスピットはセーブされず、このエラー状態は一時中断パラメータのステートから推定しなければならない。

【0362】リンクされたリストの開始アドレスまたはリンクされたリストのアドレスのオフチップの次のエントリ、整列状態のパケット転送パラメータテーブルまたはオフチップメモリへのガイドテーブルポインタによって生じたエラーは、パケット転送パラメータをリクエスト中のプロセッサのパラメータメモリの一時中断エリアにセーブしない。これはデータを転送する前にパケット転送がアボートし、実際に開始することがないからである。従って、エラーの原因を診断しようとするとき、リンクされたリストのアドレスの有効性をチェックして、一時中断されたパラメータが有効であることを保証しなければならない。

【0363】転送コントローラ80は多くの条件下でパケット転送を一時中断する。優先度のより高いパケット転送リクエストを受けると、現在のパケット転送のためにPTMINに指定されていたサイクル数がなくなる場合に、パケット転送を一時中断する。転送コントローラ80はサイクル数がPTMIN+PTMAXを越え、タイムアウトし、別のプロセッサからの同じ優先度のパケット転送リクエストがペンディング中である場合、パケット転送を一時中断する。転送コントローラ80がリクエスト中のプロセッサがこのコントローラにCOMMレジスタ120のSビットにより一時中断することを求める場合、パケット転送を一時中断する。ソースアドレス指定または宛て先アドレス指定中にメモリフォールトが生じる場合、転送コントローラ80はパケット転送を一時中断する。エラー状態が検出される場合、転送コントローラ80はパケット転送を一時中断する。また、外部メモリアクセス中に再試行が行われ、別のプロセッサか

らの同じ優先度のパケット転送リクエストが待機中であり、現在のパケット転送のためのPTMIN内に指定されたサイクル数がなくなる場合、転送コントローラ80はパケット転送を一時中断する。

【0364】これらケースの各々における一時中断機構は同一である。転送コントローラ80は現在のパケット転送パラメータおよび転送コントローラ80の内部ステートをセーブする。この情報により、パケット転送を将来続行できる。これらパラメータはリクエスト中のプロセッサのパラメータメモリの一時中断エリアにセーブされる。図15に示すように、マスタプロセッサ60がパケット転送をリクエストした場合、このエリアは16進数01010000から16進数0101007Fとなる。デジタル画像/グラフィックプロセッサ71、72、73、74の一つにリクエストされたパケット転送のための一時中断されたパケットパラメータエリアは、アドレスの16進数0101#0000から16進数0100#07F（ここで#は表3にリストしたようなデジタル画像/グラフィックプロセッサ番号に対応する）を使用する。図32および33には、これらパラメータのためのフォーマットが示されている。図32における%のマークのついたダブルワードは、現在のエンディアンのためには調節されないことに留意されたい。

【0365】一時中断条件が生じ、ペンディング中の外部メモリ行アクセスが完了した直後に、一時中断が開始する。リクエスト中のプロセッサのパラメータメモリ上の外部間バッファは空ではないが、パケット転送FIFOバッファ311の現在ステートがセーブされる。この方法により、原因にかかわらず、パケット転送は一貫して中断できる。パケット転送FIFOバッファ311の空状態は生じないので、この一時中断は高速となるようにも保証されている。従ってパケット転送FIFOバッファ311を空にするのに潜在的に低速のパケット転送を待つことなく、緊急優先度のパケット転送リクエストを急速にサービスできる。

【0366】パラメータメモリが一時中断の原因となっている場合、より高い優先度の転送コントローラ80のクロスバーの優先レベルで実行される。従って一時中断が完了するまでデジタル画像/グラフィックプロセッサ71、72、73、74またはマスタプロセッサ60は一時的な競合を経験することがある。一時中断がタイムアウト、フォールト、エラーまたはリクエスト中のプロセッサからの一時中断リクエストによるものであるときは、パケット転送の元の転送コントローラ80のクロスバーの優先度で、この一時中断が実行される。

【0367】新しいアドレスをポイントするように、リンクされたリストの開始アドレスポインタが変えられることを条件に、一時中断されたパケット転送パラメータを他の場所でコピーし、その新しい位置から再送信することができる。パケット転送オプションフィールドのバ

ケット転送ステータスビットは、ケット転送が一時中断されたことを表示するので、一時中断されたパラメータの全組は、その位置がどこであれ、再送信時にロードされる。一時中断されたパラメータの新しい開始アドレスは、偶数の128バイトの境界上で整列していなければならない。従って7つの最小位アドレスビットは、再送信すべき一時中断されているケット転送に対し00000000とならなければならない。

【0368】ケット転送一時中断エリアにセーブされた最初の8つの64ビットダブルワードは、通常のケット転送パラメータを示す。これらは、少数の例外を除き、ユーザーによってプログラムされたパラメータと同じである。最初の32ビットワードは、リンクされたリスト上の次のエントリへのアドレスでなく、一時中断されたケット転送の元のエントリアドレスを含む。Cカウントフィールドは、次元の定められた転送のための現在のCカウントを含む。ガイド転送に対し、Cカウントフィールドはガイドカウントとガイドテーブルポインタフィールドを含み、ガイドテーブルポインタフィールドはガイドテーブルポインタの現在位置を含む。更に、ケット転送オプションフィールドはパラメータが一時中断されたケット転送を表示することを示すように変えられた、そのケット転送ステータスフィールドを有する。

【0369】一時中断エリアは通常ユーザーがプログラムしたケット転送パラメータの他に8個の保留された内部ステートのダブルワードも含む。これらワードは一時中断されたケット転送が再スタートされる際に自動的にロードされるフィールドである。これらは一時中断した際のケット転送の次元内ステートに関する情報を含む。これらの値は、これらが再送信される際に残される場所で正確に一時中断されたケット転送を正しく開始できるようにセーブされる。16進数040で終了するアドレスにおけるデータワードは、現在のソースAおよびBのカウントを記憶する。これらはソースマシン320のACURRENTおよびBCURRENTからの16ビットの値である。Bのカウントはビット31~16に記憶され、Aのカウントはビット15~0に記憶される。16進数044で終了するアドレスにおけるデータワードは、現在の宛て先AおよびBのカウントを記憶する。これらは宛て先マシン340のAカウントおよびBカウントレジスタからの16ビットの値である。Bのカウントはビット31~16に記憶され、Aのカウントはビット15~0に記憶される。16進数048で終了するアドレスにおけるデータワードは、現在のソースA開始アドレスまたは現在のガイドテーブルアドレスである。一時中断されたケットサービスパラメータ内の対応するPビットは、どのデータがここに記憶されたかを表示する。元のBのカウントが0に達するまでガイドテーブルのフェッチは行われないので、ガイドケット

転送でもここにソースAの開始アドレスを記憶できるとに留意されたい。16進数04Cで終了するアドレスにおけるデータワードは、現在の宛て先Aの開始アドレスまたは現在のガイドテーブルフェッチ値であり、その識別は一時中断されたケットサービスパラメータの対応するPビットによって決定される。16進数050および056で終了するアドレスにおけるデータワードは、現在の宛て先Bの開始アドレスまたは対応するガイドテーブルフェッチ値である。

10 【0370】16進数058で終了するアドレスで開始する16ビットのダブルワードは、一時中断されたケットサービスパラメータを記憶する。一時中断されたケットサービスパラメータのフォーマットは図32および33に示されている。上部データワードのビット31~30はPビットである。ビット31における1は、ガイドテーブルからフェッチされたソースマシン320のASTARTレジスタおよびBSTARTレジスタの値を記憶する。これと異なり、これらレジスタは次元の定められた転送からの値を記憶する。同様にビット30における1は、一時中断されたケット転送パラメータがガイドテーブルからフェッチされたソースマシン320のASTARTレジスタおよびBSTARTレジスタ値を記憶し、0はこれら値が次元の定められた転送からのものであることを表示する。これらビットは一時中断されたケット転送がガイドソースまたは宛て先アドレス指定を含んでいたことを、ケット転送オプションフィールドが表示している場合に意味があるにすぎない。これとは異なり、データは次元の定められた転送からのものであると見なされる。

30 【0371】Eビット（上部データワードのビット29）は、一時中断されたケット転送がオフチップからオフチップへのケット転送であることを表示する。かかるオフチップからオフチップへのケット転送は、リクエスト中のプロセッサのパラメータメモリ内のバッファを利用する。このEビットが1であれば、一時中断されたケット転送はオフチップからオフチップへのケット転送である。このEビットが0であれば、一時中断されたケット転送は他の3つのタイプのうちの一つであり、リクエスト中のプロセッサのパラメータメモリは外部から外部へのバッファ化されたデータを記憶しない。

40 【0372】Sビット（上部データワードのビット28）は、パラメータメモリの外部間バッファがソースマシン320によってアクセスされているか、宛て先マシン340でアクセスされているかどうかを表示する。Eビットが1であって、リクエスト中のプロセッサのパラメータメモリ内の外部間でバッファ化されたデータの記憶を表示している場合にかぎり意味がある。Sビットが1である場合、ソースマシン320はケット転送が一時中断されているときに、外部間バッファにアクセス中

である。Sビットが0であれば宛て先コントローラは外部間バッファにアクセス中である。

【0373】上部データワードのビット20～16は、BUFPTRレジスタ414の値を記憶する。BUFPTRレジスタ414は、パラメータメモリの外部間バッファへのポインタである。このポインタを記憶すると、パラメータメモリの外部間バッファのステータスをリカバーすることにより、外部間パケット転送の再使用が可能となる。当然ながらこのフィールドは、一時中断パケット転送が外部間パケット転送である場合にのみ意味がある。

【0374】上部データワードのビット14～8は、BUFCOUNTレジスタ441の値を記憶する。このBUFCOUNTレジスタ441の値は、Sビットのステータスに基づいて解釈される。Sビットが1であって、パラメータメモリの外部間バッファのソースアドレス指定が一時中断していることが表示されると、BUFCOUNTレジスタ441はバッファ内に残っているバイト数を表示する。Sビットが0であり、パラメータメモリの外部間アドレスの宛て先アクセスの一時中断を表示している場合、BUFCOUNTレジスタ441はバッファ内に記憶されたバイト数を表示する。このフィールドは一時中断された外部間パケット転送にしか意味がない。上部データワードのビット3～0は、FCOUNTである。これはパケット転送FIFOバッファ311にこのとき保持されているデータのバイト数である。

【0375】一時中断されたパケットサービスパラメータの下部データワードは、2つのFIFOポインタを保持する。下部データワードのビット11～8は、FDPTRすなわち宛て先ポインタを記憶する。下部データワードのビット3～0は、FSPTTRすなわちソースポインタを記憶する。これらポインタは、一時中断点における一時中断パケット転送の再開を可能にする。

【0376】16進数060で終了するアドレスで開始する4つのデータワードは、パケット転送FIFOバッファ311の内容を記憶している。パケット転送FIFOバッファ311を空にするかわりに、このデータを記憶することにより、より高い優先度のパケット転送を早期にスタートできる。将来、より大きなバッファを使用する場合に、このような大きなパケット転送用FIFOバッファ311の内容の記憶をサポートするために、16進数070で終了するアドレスで開始する4つのデータワードが保留されていることに留意されたい。

【0377】転送コントローラ80は、多数の標準的でないパケット転送アクセスモードをサポートする。これらにはブロックライト、シリアルレジスタ転送、トランスペアレンシーおよび周辺デバイス転送が含まれる。これら非標準的パケット転送アクセスモードは、パケット転送パラメータのパケット転送オプションフィールドによりイネーブルされる。

【0378】パケット転送パラメータのパケットアクセスモードが010に等しいとき、このモードはブロック書き込みパケットアクセスモードを可能にする。転送コントローラ80は3つの異なるブロック書き込み機構、すなわち8x、3xおよびシミュレート機構をサポートする。システムハードウェアは、ブロックの書き込みが開始する時間にブロック書き込み機構を決定する。これによりソフトウェアはシステムが使用するブロック書き込みのタイプがどのタイプであるかとは無関係に、またはこれをアドレス指定されたメモリがサポートするか否かとは無関係に、ソフトウェアがブロック書き込みを利用できるようにする。しかしながらブロック書き込みモードの各々は、64ビットのバスサイズおよび8ビットの値(1～8)に対してしかサポートされていない。ブロック書き込みのための宛て先スタートアドレスはオフチップであり、64ビットに整列していなければならない。すなわち最小位の6つのアドレスビットは000000でなければならない。

【0379】転送コントローラ80によって使用されるブロック書き込みモードは、外部回路によりマルチプロセッサ集積回路100のBS[1:0]ピンに入力された値によって選択される。好ましい実施例では、ブロック書き込みは64ビットのデータバスに対してのみサポートされているので、これらバスサイズ入力ブロック書き込みおよび負荷カラーレジスタサイクル中にブロック書き込み選択として使用される。表23は、BS[1:0]により選択されたブロック書き込みモードを示す。

【0380】

【表23】

BS [1:0]		ブロック書き込みモード
0	0	シミュレートされたモード
0	1	保留予約
1	0	4x
1	1	8x

表23

【0381】種々のVRAMサイズおよびアーキテクチャのために、多数の異なるブロック書き込み方法がある。以下の表記法は、種々のブロック書き込み方法；すなわちC×L×R(ここでCはカラーレジスタ当たりの列位置の数、Lはビットによるカラーレジスタの長さ、Rはカラーレジスタの数である)を示すのに使用されている。たいていの1MビットのVRAMは、256Kビット×7ビットデータアレイを有し、4×4×1のブロック書き込みをサポートしている。これらは1つの4ビットカラーレジスタを有し、各ブロック書き込みサイクルは、4メモリアレイ平面での4つの隣接する列位置へのカラーレジスタの書き込みを制御している。

【0382】8x(倍の)ブロック書き込みは、位置当

たり8ビットに、アクセス当たり8つの行位置を(すなわち $8 \times 8 \times 1$ または $8 \times 8 \times 2$)書き込みできるVRAMと共に使用するようになっている。ソースデータの各ビットは、データバスD[63:0]上の1ビットに出力され、このサイクルでアクセスされる64個の列のうちの1つに対する8ビットVRAMカラーレジスタの書き込みをイネーブルしたりディスエーブルしたりする。各列は各VRAM内の8つのメモリアレイ平面の各々における列位置を表示している。従って1回のアクセスで64までのバイトのカラーレジスタデータを書き込み

【0383】8×ブロック書き込みをサポートするビデオRAMは、ブロック書き込みサイクル中に最小位の3つの列のアドレスを無視する。従って、データバスは64ビット(8バイト幅)であるので、ブロック書き込みサイクルは常に64バイトの境界上で開始する。64バイトに整列されていない宛て先アドレスに対し、転送コントローラ80はソースビットを整列し、書き込みされていない64バイトのアクセス内の位置に対して喪失した0を発生する。次に転送コントローラ80は、ビット

【0384】図34は、小エンディアンモードにおける8×ブロック書き込みのためのデータ再マップ化プロセスを示す。最初の8つのソースビットは宛て先のうちの最初の8バイトを制御する。これらバイトは実際は、アクセスされている8個の8平面VRAMアレイの各々における最小位の列位置である。最小位バイトはブロック書き込み中にVRAMのD0入力(および16ビットVRAMのためのD8入力)によって制御されるので、ソースビット0~7はデータバスのそれぞれのビット0、8、16、24、32、40、48および56にマップ化され、これらは、アクセスされたVRAMのD0入力(16ビットデバイスのためのD0およびD8入力)を表示する。ソースビットの残りは同様にマップ化される。転送コントローラ80は、大エンディアン8×ブロック書き込みのための同様なマッピングを行う。大エンディアンモードではソース画像のうちの最小位ビットは左側の最大ビットであり、最高位ビットは右側の最大ビットである。マッピング機構は小エンディアンモードのために使用されているものと同じであるので、ソースビット0~7はそれぞれビット63、55、47、39、31、23、15および7にマップ化されている。これらビットはVRAMアレイ内でアクセスされている最小位列位置を表示する。これらバイトは、VRAMのD0入力(16ビットデバイスのためのD0およびD8入力)によって制御されているので、ブロック書き込みが正しく作動できるように、逆の順にVRAMに接続しなければならない。

【0385】データは逆の順に書き込みされ、読み出さ

れるので、データバスを逆の順に接続しても通常の読み出しおよび書き込みに影響しない。VRAMのシリアルポートの内外にシフトされるビットは、逆の順となることをユーザーは認識しなければならない。このようにするには、適当なオペレーションを保証するため、出力デバイスまたは入力デバイスを接続する際にシリアルデータバスの順を反転しなければならないことがある。

【0386】4×ブロック書き込みモードは、アクセス当たり4列の位置、すなわち位置当たり4または8ビットのいずれかで、 $4 \times 4 \times 1$ 、 $4 \times 4 \times 4$ 、 $4 \times 8 \times 1$ 、 $4 \times 8 \times 2$ を書き込みできるVRAMと共に使用するように設計されている。 4×4 のブロック書き込みの場合、ソースデータの各ビットは、データバスD[63:0]上の2ビットに出力され、このサイクルでアクセスされる64列のうちの2つに対する4ビットVRAMカラーレジスタのうちの2つの書き込みをイネーブルまたはディスエーブルする。各列はアレイのうちの4つの平面上の列位置を表示する。各カラーレジスタは4ビット幅しかないので、8ビットのピクセルを表示するのに2つのレジスタが必要である。従って各ソースビットは8ビットの書き込みが生じるように、隣接するニブルに書き込む2つのカラーレジスタを制御しなければならない。これにより1回のアクセスでカラーレジスタデータのうちの32バイトまでを書き込みできる。

【0387】4×ブロック書き込みをサポートするVRAMは、ブロック書き込みサイクル中に最小位の2つの列アドレスを無視する。従ってブロック書き込みは常に32バイト境界上で始まる。8×ブロック書き込みと同じように、転送コントローラ80は32バイトブロック内の宛て先開始アドレスのダブルワードによって指定されたデータにソースデータを整列し、書き込まれていないダブルワードに対し、喪失した0を満たす。

【0388】図35は、小エンディアンモードにおける4×ブロック書き込みのためのデータ再マップ化プロセスを示す。最初の8つのソースビットは宛て先の最初の8つのバイトを制御する。しかしながら 4×4 ブロック書き込みでは、各データ入力では4ビットしか書き込みしないので、完全なバイトを書き込みするように2つのデータバスピンに各ソースビットをマップ化しなければならない。従ってソースビット0~7はビット0と4、8と12、16と20、24と28、32と36、40と44、48と52ならびに56と60にそれぞれマップ化される。

【0389】 $4 \times 8 \times 1$ または $4 \times 8 \times 2$ のブロック書き込みの場合、VRAMカラーレジスタは8ビット幅である。しかしながら書き込むべきバイトを選択するのに、VRAMの8個のデータ入力のうちの4つしか使用しない。従ってデータバスのうちの半分は使用せず、1回のアクセスで32バイトしか書き込みできない。しかしながら、D[63:0]上に実際に入力されるデータ

10

20

30

40

50

は同じであり、VRAMは他のどのニブルも無視するだけであることに留意されたい。大エンディアンモードでは、ソースビット0〜7は外部データバスのうちのビット63と59、55と51、47と43、39と35、31と27、23と19、15と11ならびに7と3にマップ化される。8×大エンディアンブロック書き込みと同じように、正しいオペレーションを保証するようにVRAMにデータバスを逆の順序で接続しなければならない。

【0390】ブロック書き込みをサポートしないメモリデバイスに対し、転送コントローラ80はシミュレートされたブロック書き込みモードを提供する。このモードでは、データバス上にパケット転送パラメータ内に含まれていた64ビットのカラーレジスタ値が出力され、サイクル中にアドレス指定される8バイトのうちの1つをイネーブルまたはディスエーブルするよう、CAS'列アドレスストロブピンを制御することにより、各ソースデータビットはバイト選択信号として機能する。従ってブロック書き込みは実質的にはカラーレジスタ値がフィル値となるような、値でフィルするタイプの転送に変換される。次に、宛て先アクセスは、ソースデータによって指定されるように、あるバイトへの書き込みをディスエーブルする通常のページモードの64ビットの書き込みサイクルとなる。

【0391】4×および8×ブロック書き込みサイクルを実行する前に、VRAMのカラーレジスタに正しい値をロードしなければならない。転送コントローラ80はパケット転送パラメータに含まれるカラーレジスタ値を使用するロードカラーレジスタLCRサイクルを実行することによりこれを行う。ブロック書き込みパケット転送は、より高い優先度のリクエスト、例えばVRAMカラーレジスタを変え得るホストアクセスまたは他のブロック書き込みパケット転送によってインタラプトできるので、ブロック書き込みパケット転送が再開するときはいつも、ロードカラーレジスタサイクルも実行しなければならない。従って4×または8×ブロック書き込みパケット転送が始まる時、一時中断からブロック書き込みパケット転送が再開するとき、ホストが画像システムバスを使用し、これを戻した後ブロック書き込みパケット転送が続くときはいつも、ロードカラーレジスタが始まる。アクセスされているメモリがシミュレートされたブロック書き込みを必要とする場合には、ロードカラーレジスタサイクルは実行されない。

【0392】一旦カラーラッチがロードされると、上記条件のうちの1つが発生しなければ、別のロードカラーレジスタサイクルを実行することはない。例えば8×モードでブロック書き込みが開始し、次にシミュレートされたモードと8×モードが交互に変わると、8×モードとなるたびにロードカラーレジスタが繰り返されることはない。

【0393】ブロック書き込みパケット転送のための事象シーケンスは次のとおりである。転送コントローラ80は、実行すべきロードカラーレジスタステータスコードおよび第1ブロック書き込みのアドレスを出力する。次に転送コントローラ80は、BS[1:0]ピンに入力された値を読み出す。BS[1:0]が10または11に等しければ、パケット転送パラメータ内に含まれる64ビットのカラーレジスタ値も用いてロードカラーレジスタサイクルを完了する。この後に、4×または8×ブロックサイクルを発生してパケット転送を完了する。一方、BS[1:0]が00であれば、ロードカラーレジスタサイクルはデータとしてパケット転送パラメータ内に含まれる64ビットのカラーレジスタ値およびバイト選択信号としてソースデータビットを用いる通常のページモードの書き込みとなる。ページ変更またはより高い優先度のサイクルからのインタラプトのために、新しい行アクセスが開始された場合、実行すべきロードカラーレジスタのステータスコードおよび次のブロック書き込みのアドレスを出力する工程を、次の宛て先アドレスに対して繰り返す。

【0394】ロードカラーレジスタサイクルが一旦実行されると、次のようにシーケンスが続く。転送コントローラ80がブロック書き込みステータスコードおよび実行すべき次のブロック書き込みのアドレスを出力する。BS[1:0]が10または11である場合、ソースデータビットを用いてブロック書き込みページモードサイクルが完了される。新しい行アクセスが開始されると、ブロック書き込みステータスコードおよび実行すべき次のブロック書き込みのアドレスを出力するステップが繰り返される。他方、BS[1:0]が00であれば、ブロック書き込みサイクルはパケット転送内に含まれる64ビットのカラーレジスタ値をデータとして使用し、ソースデータビットをバイト選択ビットとして使用する、通常のページモード書き込みとなる。新しい行アクセスが開始されると、ブロック書き込みステータスコードおよび実行すべき次のブロック書き込みのアドレスを出力するステップが次の宛て先アドレスに対して繰り返される。

【0395】転送コントローラ80は実際の4×または8×ブロック書き込みを常に実行するように試みる。従ってシミュレートされたブロック書き込みモード中に生じる通常の書き込みサイクルは、常にロードカラーレジスタまたはブロック書き込みステータスコードを有する。

【0396】パケット転送オプションフィールド内のパケットアクセスモードが、シリアルレジスタ転送モードを選択する011であれば、転送コントローラ80のデータ転送モードがディスエーブルされる。転送コントローラ80はアドレスおよびCAS'を出力するだけである。転送コントローラは更に行時間にTRG'、W'お

よびDSF [1:0] ビンもドライブし、読み出し転送または書き込み転送のVRAMモードのいずれかを選択する。これらオペレーションのいずれかを実行する際、D [63:0] ビンは行時間に16進数FFFFFFFFFFFFFFFFのマスク値を出力して、VRAM転送マスクをディスエーブルする。

【0397】メモリからレジスタへの転送である読み出し転送を実行するためのアドレスを発生するのにソースパラメータが使用され、レジスタからメモリへの転送である書き込み転送を実行するのに宛て先メモリが使用される。

【0398】ソースおよび宛て先の双方によって実行される各アクセスは、1回の行アクセスである。転送コントローラ80を介するデータの転送は行われないので、宛て先アクセス前にすべてのソースアクセスが実行される。各ソースアクセスはVRAMの行をVRAMシフトレジスタ内に転送させる。実際にはソースパラメータは通常整列したアドレス、1のAカウンタ、0のBカウンタおよび0のCカウンタでセットアップされる。従って1回の転送しか実行されない。読み出し転送に対してはすべての反転CASラインはアクティブであるので、ソースバスサイズ以下のAカウンタをセッティングする結果、1回の転送が行われる。従ってほとんどの状況に対して1のソースAカウンタが理想的である。

【0399】各宛て先アクセスは、VRAMシフトレジスタのデータをVRAMメモリアレイの行内に転送させる。通常、各宛て先アクセスは次の逐次行アドレスに対する整列された転送となっている。これは、整列された開始アドレス、1のAカウンタ、行数-1のBカウンタおよびVRAM行アドレスピッチに等しいBピッチで宛て先パラメータをセットアップすることによって行われる。書き込み転送中、すべての反転CASラインが附勢されるので、宛て先バスサイズ以下のAカウンタをセットする結果、ライン当たり1回の転送が行われる。従って1の宛て先Aカウンタはほとんどの状況に対して理想的である。

【0400】シフトレジスタ転送パケット転送は、VRAMシフトレジスタの内容を変え得る優先度のより高いリクエストによりインタラプトされ得るので、VRAMシフトレジスタが損なわれる可能性がある場合はいつも、読み出し転送サイクルを実行する必要がある。従って、シフトレジスタ転送パケット転送が開始するとき、シフトレジスタ転送パケット転送が一時中断された後に再開するとき、更にホストが使用され、画像システムバスに復帰した後にシフトレジスタ転送パケット転送が続くときはいつも、シフトレジスタ転送パケット転送ノソースオペレーション、すなわち読み出し転送が行われる。

【0401】図37にシリアルレジスタ転送パケット転送の簡単な例が示されている。行0は所望のパターンに

セットされているものとする。パケット転送はこのパターンは行2、4、6、8および10にコピーし、ストライプ状効果を発生するためのものである。

【0402】パケット転送オプションフィールド内のパケット転送アクセスモードビットを、1XXにセットすることにより、トランスペアレンシーモードをイネーブルする。トランスペアレンシーモードのうちの1つを指定することによりソース上でのトランスペアレンシーオペレーションがイネーブルされる。ソースデータはパケット転送パラメータ内に指定された64ビットのトランスペアレンシーモード値と比較される。トランスペアレンシーは8、16、32または64ビットデータサイズとして指定できる。1回の64ビットの比較、2回の32ビットの比較、4回の64ビットの比較または8回の8ビットの比較を行う。比較の結果が真であれば、トランスペアレンシーおよびバイト書き込み回路353（図38に示される）が対応するバイトストローブをディスエーブルし、宛て先バイトの書き込みを防止する。トランスペアレンシーはオフチップの宛て先に対してしかサポートされていない。オンチップの宛て先に対するトランスペアレンシーを指定すると、エラー条件によりパケット転送が一時中断される。

【0403】ソースデータが宛て先および外部バスサイズに整列された後、トランスペアレンシーおよびバイト書き込み回路353（図38に示されている）によりトランスペアレンシー検出が行われる。バスサイズが64ビット未満であっても、データのうちのすべての8バイトがトランスペアレンシー値の対応する8バイトと比較される。8回の比較は、トランスペアレンシーデータサイズに従ってグループ分けされる。グループ内の比較されるバイトがすべて一致すると、そのグループに関連するバイトストローブCAS' 信号がディスエーブルされ、そのグループ内のバイトのいずれかへの書き込みが防止される。

【0404】図37a、37b、37cおよび37dは、それぞれ64ビット、32ビット、16ビットおよび8ビットのトランスペアレンシーデータサイズに対してどのように比較を行うかを示している。&記号はグループを形成するのに、どのバイトの比較をAND演算するかを示している。図37aが示すように、64ビットのトランスペアレンシーサイズは1回の64ビットの比較を行う。ソースデータおよびトランスペアレンシー値が等しければCAS' ストロブのすべてがディスエーブルされる。それ以外の場合、8バイトのすべてが書き込まれる。図37bは、32ビットのトランスペアレンシーサイズで2回の32ビットの比較が行われ、それぞれCAS' [7:4] およびCAS' [3:0] を制御することを示している。図37cは、16ビットのトランスペアレンシーサイズにおいて、4回の16ビットの比較が行われ、それぞれCAS' [7:6]、CAS'

[5:4]、CAS' [3:2] および CAS' [1:0] を制御することを示している。図37dは8ビットのトランスペアレンシーサイズにて8回の8ビットの比較を別々に行い、CAS' [7]、CAS' [6]、CAS' [5]、CAS' [4]、CAS' [3]、CAS' [2]、CAS' [3:2]、CAS' [1:0] を制御することを示している。CAS' [7:0] ストロープは、オペレーションのエンディアンにかかわらず、データバス上の同一ビットにより常時識別される。

【0405】外部バスへの整列後、トランスペアレンシーの比較が行われる。従って外部バスサイズが32ビットであれば、64ビットのトランスペアレンシーモードが選択されていても、データは常にトランスペアレンシー値のうちのビット31~0（大エンディアンモードではビット63~32）と比較される。このトランスペアレンシー機構は整数の比較グループに分割できる現在のバスサイズで作動するようになっている。従ってバスサイズは常にトランスペアレンシーサイズ以上でなければならない。

【0406】図38は、宛て先マルチプレクサおよび整列ロジック350の一部である、トランスペアレンシーおよびバイト書き込みロジック353の構造を示す。トランスペアレンシーレジスタ601は、パケット転送パラメータからの64ビットのトランスペアレンシー値を記憶する。一連の8ビットのコンパレータ611、612、613、614、615、616、617および618は、トランスペアレンシーレジスタ601内に記憶されたトランスペアレンシー値のバイトの個々のビットと、宛て先マルチプレクサ351からの整列された宛て先データのバイトの対応するビットとを比較する。各コンパレータ611、612、613、614、615、616、617、618は、対応するビットが同じであるかどうかを表示するバイトの等しい信号を発生する。ハーフワードANDゲート621、622、623および624の一组は、バイト0と1、バイト2と3、バイト4と5、並びにバイト6と7に対する対応するハーフワードの等しい信号をそれぞれ形成する。2つのワードANDゲート631および632は、ハーフワードANDゲート621、622、623および624に接続されており、バイト0~3およびバイト4~7に対する対応するワードの等しい信号を形成する。最後に、ダブルワードのANDゲート641はワードゲート631および632に接続されており、すべての64ビットに対するダブルワードの等しい信号を形成する。

【0407】マルチプレクサ645は8個のバイトの等しい信号、4つのハーフワードの等しい信号、2つのワードの等しい信号およびダブルワードの等しい信号を受ける。このマルチプレクサ645はパケット転送オプションフィールドのパケット転送アクセスモードのビット17~16に従って選択されたトランスペアレンシーサ

イズの表示も受ける。これらビットは表19に示されるようにコード化されている。マルチプレクサ645の出力バイト書き込みストロープは、それぞれの入力およびトランスペアレンシーサイズに基づき、反転CAS信号を実際に発生する。8ビットのトランスペアレンシーサイズが選択される場合、バイトの等しい信号がそれぞれのバイト書き込みストロープを制御する。トランスペアレンシーカラー値と、対応するデータバイトが等しいことの表示は、バイト書き込みストロープを禁止する。従ってメモリにはデータは書き込まれない。16ビットのトランスペアレンシーサイズが選択されると、各ハーフワードの等しい信号は2つの対応するバイト書き込みストロープを制御し、よって、各ハーフワードの等しい信号は2バイトの書き込みストロープを制御する。32ビットのトランスペアレンシーサイズが制御されると、各ワードの等しい信号は4つの対応するバイト書き込みストロープを制御する。64ビットのトランスペアレンシーサイズが選択されると、8つのバイト書き込みストロープのすべてがダブルワードの等しい信号によって制御される。従って、選択されたトランスペアレンシーサイズに基づき、トランスペアレンシーサイズに等しいデータ部分とトランスペアレンシーカラー値との同一性により、データの宛て先への書き込みがアボートされる。この機能は多くのグラフィックアプリケーションで極めて有効である。

【0408】パケット転送オプションのパケットアクセスモードが001に等しいときに、周辺デバイスモードが選択されると、周辺デバイス転送が実行される。このモードは周辺デバイスが転送コントローラ80のメモリコントローラを活用し、マルチプロセッサ集積回路100の外部のメモリとの間における読み出しまたは書き込みを可能にする。周辺デバイス転送が行われると、転送コントローラ80はメモリアドレスおよび制御ラインをドライブするが、データを読み出ししたり、ドライブ出力させることはない。これにより、周辺デバイス、例えば画像システムバスに接続されているホストシステム1は、データの書き込みまたは読み出しを可能にする。周辺転送の読み出しまたは書き込み方向は、パケット転送パラメータをプログラムする方法により決定される。

【0409】メモリから周辺デバイスへの読み出しを行う周辺読み出し転送は、周辺デバイスが必要とするメモリデータにアクセスするよう、パケット転送のソースパラメータをプログラムすることによって発生される。ソースアクセスモードは、次元の定められた転送またはガイド転送のいずれかにできるが、値でフィルする転送にはできない。宛て先転送は宛て先転送モードを000フィールドにセットし、宛て先Aカウントを0にセットすることによりディスエーブルしなければならない。

【0410】周辺デバイスからメモリへの書き込みを行う周辺書き込み転送は、周辺デバイスが書き込みを必要

とするメモリエリアにアクセスするよう、宛て先パラメータをプログラムすることによって発生される。宛て先アクセスモードは次元の定められた転送またはガイド転送のいずれかにできる。ソース転送はソース転送モードを000にセットし、ソースAカウンタを0にセットすることによりディスエーブルしなければならない。バイトの宛て先番号がバイトのソース番号を越えたとしても、この例ではパケット転送エラーは生じない。

【0411】周辺デバイスパケット転送リクエストは、任意の優先度のプロセッサにより送信でき、通常の優先度決定方法を用いてサービスされる。しかしながら通常のオペレーションはデータの読み出し、書き込みを望む際に、周辺デバイスに転送を開始させるようになってい10
る。このような開始は、外部インタラプトを用いることによって行われる。周辺デバイスがサービスを必要とする際、周辺デバイスはマルチプロセッサ集積回路100の外部のインタラプト入力の一つにより、マルチプロセッサ60をインタラプトできる。マスタプロセッサ60は次に、インタラプトサービスルーチンの一部として周辺デバイスパケット転送を送ることができる。サービス20
ルーチン内でインタラプトがディスエーブルされれば、この転送は緊急優先度を送ることにより、デジタル画像/グラフィックプロセッサ71、72、73、74のキャッシュリクエストよりも高い優先度を与えることができる。

【0412】周辺デバイスがマスタプロセッサ60を一旦インタラプトすると、周辺デバイスは転送コントローラ80がデータの読み出し、書き込みをできる前に、パケット転送を開始するまで待機しなければならない。周辺デバイスの転送の開始は、行時間におけるステータス30
【4:0】上に出力される特別サイクルタイプのコードによって信号が送られる。読み出しに対しては値00100が使用され、書き込みに対しては値00101が使用される。周辺デバイス転送の開始点および終了点は、常にLASTPAGEレジスタ360を無効にし、行アクセスを強制的に行わせる。周辺デバイスはデータ転送をいつ行うかを決定するよう、STATUS【4:0】をモニタしなければならない。転送コントローラ80により発生されるメモリサイクルのバスサイズの種類、列タイミング等は、サイクルの開始点におけるメモリ識40

別入力バスサイズ選択BS【1:0】、列タイミング選択CT【1:0】等によって選択されるものに対応する。次に周辺デバイスは、CAS'【7:0】、CLKOUT等を用いる、その後の列アクセスにそのデータ転送を同期化できる。

【0413】メモリのアドレス指定はパケット転送パラメータによって行われるので、転送周辺デバイスでは転送コントローラ80によってアクセスされる順で、データを送受信するように準備が整っていなければならない。周辺デバイスは転送コントローラ80の転送レートに合致したり、転送を低速にするよう、待機ステートを挿入できるようになっていなければならない。

【0414】周辺デバイスは転送のために画像システムバスを使用しているので、周辺デバイス転送が開始するまでバスをドライブしてはならない。これは周辺デバイスとマルチプロセッサ集積回路100のデータバスとの間にトランシーバを置き、周辺デバイス転送中にこれらをイネーブルするだけで達成できる。マルチプロセッサ集積回路100は転送中にそのデータバスを高インピーダンスとし、外部トランシーバが設けられていれば、これをディスエーブルするよう、反転DBENをハイレベルで非アクティブとなるようにドライブする。

【0415】転送コントローラ80は、大エンディアンフォーマットまたは小エンディアンフォーマットのいずれかでデータにアクセスできる。このエンディアンモードは、バイトをアクセスする方法を選択する。小エンディアンフォーマットではバイト0はワード内の最も右側のバイトであり、その後続くバイトは左に向かって番号がつけられる。大エンディアンフォーマットではバイト0はワード内の最も左側のバイトであり、その後続くバイト2は右側に番号がつけられる。

【0416】アドレスの最小位の3ビットおよび転送すべきバイト数は、有効データバイトの位置を決定する。表24aおよび24bは、小エンディアンモードで64ビットバス転送のためのバイト位置を示す。表24aおよび24bではVは有効バイト位置を示し、0は無効バイトを示す。点線はそのオペレーションを実行できないことを示す。

【0417】

【表24】

127
32ビットの
アドレスビット

バイト数

	1バイト	2バイト	3バイト	4バイト
0 0 0	0000000V	000000VV	00000VVV	0000VVVV
0 0 1	000000V0	000000V0	0000VVV0	000VVVV0
0 1 0	000000V0	0000VV00	000VVVV0	00VVVV00
0 1 1	0000V000	000VV000	00VVVV00	0VVVV000
1 0 0	000V0000	00VV0000	0VVV0000	VVVV0000
1 0 1	00V00000	0VV00000	VVV00000	--
1 1 0	0V000000	VV000000	--	--
1 1 1	V0000000	--	--	--

表24a

32ビットの
アドレスビット

バイト数

	5バイト	6バイト	7バイト	8バイト
0 0 0	000VVVVV	00VVVVVV	0VVVVVVV	VVVVVVVV
0 0 1	00VVVVV0	0VVVVVV0	VVVVVVV0	--
0 1 0	0VVVVVV0	VVVVVVV0	--	--
0 1 1	VVVVV000	--	--	--
1 0 0	--	--	--	--
1 0 1	--	--	--	--
1 1 0	--	--	--	--
1 1 1	--	--	--	--

表24b

【0418】表25aおよび25bは、大エンディアン 20* 【0419】
モードにおける64ビットバス転送のためのバイト位置 【表25】
を示す。

32ビットの
アドレスビット

バイト数

	1バイト	2バイト	3バイト	4バイト
0 0 0	V0000000	VV000000	VVV00000	VVVV0000
0 0 1	0V000000	0VV00000	0VVV0000	0VVVV000
0 1 0	00V00000	00VV0000	00VVV000	00VVVV00
0 1 1	000V0000	000VV000	000VVVV0	000VVVV0
1 0 0	0000V000	0000VV00	0000VVV0	0000VVVV
1 0 1	00000V00	00000VV0	00000VVV	--
1 1 0	000000V0	000000VV	--	--
1 1 1	0000000V	--	--	--

表25a

32ビットの
アドレスビット

バイト数

	5バイト	6バイト	7バイト	8バイト
0 0 0	VVVVV000	VVVVVV00	VVVVVVV0	VVVVVVVV
0 0 1	0VVVVV00	0VVVVVV0	0VVVVVVV	--
0 1 0	00VVVVV0	00VVVVVV	--	--
0 1 1	000VVVVV	--	--	--
1 0 0	--	--	--	--
1 0 1	--	--	--	--
1 1 0	--	--	--	--
1 1 1	--	--	--	--

表25b

【0420】外部バスが32ビットに制限されていると
き、データの転送にはバスのうちの最小位の4バイトし
か使用しない。このことは、小エンディアン転送ではD
[31:0]が使用され、大エンディアン転送に対して
はD[63:32]が使用されることを意味している。
表26には、小エンディアンのためのアドレスの最小位

の2ビットに基づくバイト位置が示されている。Xは無
視する64ビットバスのバイトを表示し、点線は実行で
きない転送を示している。

【0421】

【表26】

129

130

2つの最小位の アドレスビット		バイト数			
		1バイト	2バイト	3バイト	4バイト
0	0	XXXX000V	XXXX00VV	XXXX0VVV	XXXXVVVV
0	1	XXXX00V0	XXXX0VV0	XXXXVVV0	---
1	0	XXXX0V00	XXXXVV00	---	---
1	1	XXXXV000	---	---	---

表26

【0422】表27には、大エンディアンのためのアドレスの最小位の2ビットに基づくバイト位置が示されている。
*【0423】
【表27】

*10

2つの最小位の アドレスビット		バイト数			
		1バイト	2バイト	3バイト	4バイト
0	0	000VXXXX	00VVXXXX	0VVVXXXX	VVVVXXXX
0	1	00V0XXXX	0VV0XXXX	VVV0XXXX	---
1	0	0V00XXXX	VV00XXXX	---	---
1	1	V000XXXX	---	---	---

表27

【0424】外部バスが16ビットに制限されていると、データの転送のために小エンディアンのための最小位の2バイトD[15:0]または大エンディアンのためのD[63:48]が使用される。表28は、アドレスの最小位バイトに基づく小エンディアンモードのためのバイト位置を示す。

【0425】

【表28】

最小位の アドレスビット		バイト数	
		1バイト	2バイト
0		XXXXXX0V	XXXXXXVV
1		XXXXXXV0	---

表28

【0426】表29は、アドレスの最小位ビットに基づく小エンディアンモードのためのバイト位置を示す。

【0427】

【表29】

最小位の アドレスビット		バイト数	
		1バイト	2バイト
0		V0XXXXXX	VVXXXXXX
1		0VXXXXXX	---

表29

【0428】外部バスが8ビット用に構成されているとき、データ転送のために小エンディアン用の最小位のバイトD[7:0]または大エンディアン用D[63:56]だけが使用される。

【0429】内部クロスバー上で32ビットの外部データを転送する際、転送コントローラ80は表24a、24b、25aおよび25bからの1バイト、2バイト、

3バイトおよび4バイトのバイト位置を用いて、通常64ビットのアクセスを実行する。内部クロスバー上で16ビットの外部データを転送する際、転送コントローラ80は表24a、24b、25aおよび25bからの1バイトおよび2バイトのバイト位置を用いて、通常の64ビットアクセスを実行する。

【0430】反転UTIME入力を用いてリセット時にマルチプロセッサ集積回路100のエンディアンモードが選択される。マルチプロセッサ集積回路100は反転リセット入力上の立ち上がりエッジの前のクロックサイクルにおけるUTIME'の値をサンプリングし、合致する。UTIME'がリセットの終了時に低レベル

(0)にサンプリングされた場合、マルチプロセッサ集積回路100は次のハードウェアのリセットが生じるまで、大エンディアンモードで動作する。UTIME'が高レベル(1)でサンプリングされると、マルチプロセッサ集積回路100は小エンディアンモードで動作する。

【0431】パケット転送パラメータはこれまで述べたように、ワード(32ビット)レベルのみにおいて、エンディアンと独立している。転送コントローラ80はパケット転送パラメータをダブルワード(64ビット)転送として常にフェッチし、記憶する。転送コントローラ80は選択されたエンディアンに従って32ビットのワードをスワップする。ワード内の16ビットの量、例えばAカウントおよびBカウントは、これら値を含むフィールドが単一の32ビットの量とみなされるので、エンディアンに従ってスワップされることはない。同様に、64ビットのトランスペアレンシーワードおよび64ビットのカラーレジスタ値のフィールドは、常に単一の64ビット量として取り扱われ、そのバイトはエンディアンに従ってスワップされることはない。

【0432】ローカルメモリの読み出しおよび書き込みサイクルは、メモリとプロセッサ集積回路100との間でデータとインストラクションとを転送するのに使用される。これらサイクルはパケット転送、キャッシュリクエストまたは転送コントローラ80に対するダイレクト外部アクセスリクエストの結果として生じ得る。読み出しサイクルはデータをメモリからマルチプロセッサ集積回路100へ転送する。転送コントローラ80はこのサイクルの開始点でSTATUS[4:0]上に00000を出力し、読み出しが行われていることを表示する。サイクル中WE'は高レベルで非アクティブに保持され、TRG'はRAS'の降下後に低レベルにドライブされ、メモリ出力ドライバをイネーブルし、DDIN'はデータトランシーバがマルチプロセッサ集積回路100の内部をドライブするように、このサイクル中に低レベルでアクティブとなる。転送コントローラ80はD[63:0]をメモリから駆動できるように高インピーダンスにスイッチングし、適当な熱ステートの間に入力データをラッチする。転送コントローラ80は常に64ビットのダブルワードを読み出し、次に適当なデータバイトとを抽出し、整列する。従って64ビット未満のバスサイズに対して、無効バイトを放棄する。

【0433】図39は、高品位テレビシステムにおける、本発明に係わるマルチプロセッサ集積回路100の使用法を示す。図39は、高品位テレビ信号の3つのソースを示している。これら信号としては、放送テレビ信号、コンパクトディスクのリードオンリーメモリ信号およびケーブルテレビ信号がある。

【0434】アンテナ801は高品位テレビ信号を含む放送用無線周波数信号を受信し、テレビチューナー802は特定の無線周波数信号を選択する同調受信機と、無線周波数信号上にエンコードされた画像データを抽出する複合器と、アナログ/デジタルコンバータを含む。従ってテレビチューナー802は、高品位テレビ画像に対応するデジタル信号を発生する。これらデジタル信号はバッファ803に一時的に記憶されるようになってい

る。

【0435】コンパクトディスクリードオンリーメモリ(CDROM)プレーヤー811は、コンパクトディスク上に永久記録されたデータを読み出す。これらデータは、所望の高品位テレビプログラムに対応した画像データを含む。コンパクトディスクリードオンリーメモリプレーヤー811は、コンパクトディスクから読み出したデジタルデータを一時記憶のためバッファ812へ供給する。バッファ812は、画像システムバスにも接続されている。

【0436】ケーブルシステムボックス821は、ケーブルシステムに双方向に接続する。このような双方向の接続によりケーブルシステムからユーザーに高品位テレビ信号の送信を行い、更にユーザーからケーブルシステ

ムへリクエスト、質問等の送信を行うことができるようになってい

る。ケーブルシステムはユーザーへのデジタル送信を利用したり、上記無線周波数放送に類似する無線周波数送信を利用したりできる。ケーブルシステムボックス821は、デジタル画像データを一時記憶できるよう、バッファ822に供給するための必要な変換回路を含む。ここでバッファ822画像システムバスにも接続されていることに留意されたい。高品位テレビは送信モードに拘わらずデータ圧縮フォーマットで送信する可能性がかなり高い。マイクロプロセッサ集積回路100は、圧縮されたデータを受信し、このデータを個々のテレビフレームにデコンプレス(圧縮解凍)し、フレームデータをビデオランダムアクセスメモリ6に供給するようにプログラムされている。先に述べたように、このデータはビデオランダムアクセスメモリ6からリコールされ、ビデオパレット7へ供給される。ビデオパレット7は、適当なビデオ信号を発生し、ビデオディスプレイ8をドライブする。画像データをディスプレイに供給する際に、画像データをデコンプレスする方法は、マイクロプロセッサ集積回路100内および画像システムバスに沿った多数のデータ移動を伴う。転送コントローラ80はマスタプロセッサ60およびデジタル画像/グラフィックプロセッサ71、72、73および74からのパケット転送用リクエストに応答し、このデータ移動を制御し、更にメモリリフレッシュのような他の画像システムバスの使用と、このデータ移動とを調和させる。

【0437】図40は、本発明の別のシステムの実施例を示す。図40では、マルチプロセッサの集積回路101はマスタプロセッサ60と単一のデジタル画像/グラフィックプロセッサ71を含む。マルチプロセッサ集積回路101はマルチプロセッサ集積回路100よりも狭いシリコン基板面積しか必要としないので、より安価に製造できる。マルチプロセッサ集積回路101は、マルチプロセッサ集積回路100の製造に対して先に述べた技術と同じ技術を用いて製造される。各デジタル画像/グラフィックプロセッサの幅は、対応するメモリおよびクロスバー50の関連する部分の幅と一致するので、マルチプロセッサ集積回路100をデジタル画像/グラフィックプロセッサ71と72との間でカットし、マルチプロセッサ集積回路101を得ることができる。4つのデジタル画像/グラフィックプロセッサの処理容量が不要の場合には、アプリケーションのためマルチプロセッサ集積回路101を用いることができる。

【0438】図42では、マルチプロセッサ集積回路101がカラーファクシミリ装置の一部として示されている。モデム1301は送受信のための電話回線に双方に結合されている。モデム1301は、バッファ1302とも通信し、このバッファは画像システムバスに更に結合されている。モデム1301は電話回線を介してファクシミリ信号を受信し、モデム1301はこれら信号を

10

20

30

40

50

復調し、復調信号は次にバッファ 1302 に一時的に記憶される。転送コントローラ 80 はデジタル画像／グラフィックプロセッサ 71 によって処理できるよう、データメモリ 22、23、24 へデータを転送することにより、バッファ 1302 にサービスする。デジタル画像／グラフィックプロセッサ 71 が、入進データの前に位置づけることができない場合、転送するコントローラ 80 はこのデータの場合 1302 からメモリ 9 へ転送することもできる。デジタル画像／グラフィックプロセッサ 71 は入進ファクシミリ画像データを処理する。この処理では、画像デコンプレッション、ノイズ低減、誤り訂正、カラーベース補正等を行ってもよい。一旦処理した場合、転送コントローラ 80 は画像データをデータメモリ 22、23、24 からビデオランダムアクセスメモリ (VRAM) 1303 へ転送する。プリンタコントローラ 1304 は、フレームコントローラ 90 の制御により、画像データをリコールし、これをカラープリンタ 1305 へ供給し、このプリンタはハードコピーを作成する。

【0439】図 40 の装置はカラーファクシミリを送ることもできる。撮像デバイス 3 はソース原画をスキャンする。撮像デバイス 3 はフレームコントローラ 90 の制御により作動している画像キャプチャコントローラ 4 へ、生の画像データを供給する。この画像データは、ビデオランダムアクセスメモリ 1303 に記憶される。図 40 に示された実施例は、別個のビデオランダムアクセスメモリを利用している図 1 の実施例と対照的に、画像キャプチャと画像ディスプレイの双方のためにビデオランダムアクセスメモリ 1303 を共用している。転送コントローラ 80 は、この画像データをデータメモリ 22、23、24 へ転送する。次にデジタル画像／グラフィックプロセッサ 71 は、データ圧縮、誤り訂正冗長性、カラーベース補正等のために画像データを処理する。この処理されたデータはファクシミリ転送をサポートするのに必要のように、転送コントローラ 80 によってバッファ 1303 へ転送される。転送コントローラ 80 は相対的データレートに応じてバッファ 1302 への転送前に一時的にメモリ 9 にデータを記憶する。バッファ 1302 内のこの画像データは、モデム 1301 により変調され、電話回線を通して送信される。

【0440】撮像デバイスとカラープリンタとが同じシステム内に設けられているので、このシステムはカラー複写機としても作動することに留意されたい。この場合、データ圧縮とデコンプレッションは不要である。しかしながらノイズ低減およびカラーベース補正のためには、まだデジタル画像／グラフィックプロセッサ 71 が有効である。コピーが原画と異なるカラーを有するように、色を注意深くずらすように、デジタル画像／グラフィックプロセッサ 71 をプログラムすることも可能である。フォールスカラーリングとして知られているこの技

術は、データのダイナミックレンジを利用可能なプリントカラーのダイナミックレンジに合わせるのに有効である。

【0441】以上の説明に関して更に以下の項を開示する。

(1) 複数の対応するアドレスにデータを記憶するメモリと、パケット転送リクエストと、パケット転送パラメータとを受ける動作をする制御回路であって、該パケット転送パラメータが、スタートアドレスと、ある数のガイドテーブルエントリと、テーブルポインタとを含む、前記制御回路と、前記数のガイドテーブルエントリを含むガイドテーブルであって、それぞれのガイドテーブルエントリが、アドレス値およびアドレスのブロックを定義するディメンション値を含み、前記テーブルポインタが最初に前記ガイドテーブル内の第 1 ガイドテーブルエントリをポイントする、前記ガイドテーブルと、前記制御回路に結合せしめられたアドレス発生回路であって、該アドレス発生回路が、前記パケット転送パラメータから、前記スタートアドレスと、前記数のガイドテーブルエントリと、前記テーブルポインタとを受ける動作をし、前記アドレス発生回路が、前記ガイドテーブルエントリに対応するメモリアクセス用のアドレスのブロックの集合を、前記スタートアドレスと、前記テーブルポインタによりポイントされた前記ガイドテーブルエントリの前記アドレス値と、の所定の組合せから、ブロックスタートアドレスであって、該第 1 ブロックスタートアドレスが前記スタートアドレスである、前記ブロックスタートアドレスを形成し、該ブロックスタートアドレスと、前記テーブルポインタによりポイントされた前記ガイドテーブルエントリの前記ディメンション値と、からアドレスのブロックを形成し、アドレスの該ブロックにメモリアクセスを行い、もしガイドテーブルエントリの前記数よりも少ないメモリアクセスしか行われ終わっていなければ、前記メモリアクセスに続いて、前記ガイドテーブルの次のエントリをポイントするように前記テーブルポインタを更新する、ことによって形成する、前記アドレス発生回路と、を含む、データ処理装置。

【0442】(2) 前記アドレス発生回路が、前記アドレス値を前記前のブロックスタートアドレスに加算することにより、前記スタートアドレスと、前記テーブルポインタによりポイントされた前記ガイドテーブルエントリの前記アドレス値と、の前記所定の組合せを形成する、第 1 項記載のデータ処理装置。

【0443】(3) 前記アドレス発生回路が、前記テーブルポインタによりポイントされた前記ガイドテーブル値を前記スタートアドレスに加算することにより、前記スタートアドレスと、前記テーブルポインタによりポイントされた前記ガイドテーブルエントリの前記アドレス値と、の前記所定の組合せを形成する、第 1 項記載のデータ処理装置。

【0444】(4)それぞれのガイドテーブルエントリの前記ディメンション値が、画素のアレイの水平ディメンション値および垂直ディメンション値を含む、第1項記載のデータ処理装置。

【0445】(5)前記メモリアクセスが、アドレスの前記ブロックからのメモリリードから成る、第1項記載のデータ処理装置。

【0446】(6)前記メモリアクセスが、アドレスの前記ブロックへのメモリライトから成る、第1項記載のデータ処理装置。

【0447】(7)単一半導体チップ上に、メモリと、パケット転送リクエストを発生する回路を有するデータプロセッサと、前記メモリと、前記データプロセッサと、に接続されたデータ転送制御装置であって、該データ転送制御装置が、パケット転送リクエストと、パケット転送パラメータとを受ける動作をする制御回路であって、該パケット転送パラメータが、スタートアドレスと、ある数のガイドテーブルエントリと、テーブルポインタとを含む、前記制御回路と、前記数のガイドテーブルエントリを含むガイドテーブルであって、それぞれのガイドテーブルエントリが、アドレス値およびアドレスのブロックを定義するディメンション値を含み、前記テーブルポインタが最初に前記ガイドテーブル内の第1ガイドテーブルエントリをポイントする、前記ガイドテーブルと、前記制御回路に結合せしめられたアドレス発生回路であって、該アドレス発生回路が、前記パケット転送パラメータから、前記スタートアドレスと、前記数のガイドテーブルエントリと、前記テーブルポインタとを受ける動作をし、前記アドレス発生回路が、前記ガイドテーブルエントリに対応するメモリアクセス用のアドレスのブロックの集合を、前記スタートアドレスと、前記テーブルポインタによりポイントされた前記ガイドテーブルエントリの前記アドレス値と、の所定の組合せから、ブロックスタートアドレスであって、該第1ブロックスタートアドレスが前記スタートアドレスである、前記ブロックスタートアドレスを形成し、該ブロックスタートアドレスと、前記テーブルポインタによりポイントされた前記ガイドテーブルエントリの前記ディメンション値と、からアドレスのブロックを形成し、アドレスの該ブロックにメモリアクセスを行い、もしガイドテーブルエントリの前記数よりも少ないメモリアクセスしか行われ終わっていなければ、前記メモリアクセスに続いて、前記ガイドテーブルの次のエントリをポイントするように前記テーブルポインタを更新する、ことによって形成する、前記アドレス発生回路と、を含む、前記データ転送制御装置と、を備えた、集積回路。

【0448】(8)前記アドレス発生回路が、前記アドレス値を前記前のブロックスタートアドレスに加算することにより、前記スタートアドレスと、前記テーブルポインタによりポイントされた前記ガイドテーブルエント

リの前記アドレス値と、の前記所定の組合せを形成する、第7項記載の集積回路。

【0449】(9)前記アドレス発生回路が、前記テーブルポインタによりポイントされた前記ガイドテーブル値を前記スタートアドレスに加算することにより、前記スタートアドレスと、前記テーブルポインタによりポイントされた前記ガイドテーブルエントリの前記アドレス値と、の前記所定の組合せを形成する、第7項記載の集積回路。

10 【0450】(10)それぞれのガイドテーブルエントリの前記ディメンション値が、画素のアレイの水平ディメンション値および垂直ディメンション値を含む、第7項記載の集積回路。

【0451】(11)前記メモリアクセスが、アドレスの前記ブロックからのメモリリードから成る、第7項記載の集積回路。

【0452】(12)前記メモリアクセスが、アドレスの前記ブロックへのメモリライトから成る、第7項記載の集積回路。

20 【0453】(13)前記単一半導体チップ上に含まれない外部メモリにアクセスする動作をする外部メモリインタフェースをさらに含み、前記アドレス発生回路が、前記メモリおよび前記外部メモリの双方を含むアドレスを発生する、第7項記載の集積回路。

30 【0454】(14)複数の位置に情報を記憶するステップと、パケットリクエストを受けてスタートアドレスおよび制御信号を供給するステップと、該スタートアドレス、制御信号、およびパッチ値にตอบสนองしてパッチアドレスを発生するステップと、パッチ情報をデコードしてディメンション値およびパッチ値を供給するステップと、前記スタートアドレスと、前記ディメンション値と、前記パッチ値とにตอบสนองしてパッチアドレスを発生するステップと、を含む、データ処理装置を操作する方法。

40 【0455】(15)複数の対応するアドレスにデータを記憶するメモリと、パケット転送リクエストと、パケット転送パラメータとを受ける動作をする制御回路であって、該パケット転送パラメータが、スタートアドレスと、アドレスのブロックを定義するディメンション値と、ある数のガイドテーブルエントリと、テーブルポインタとを含む、前記制御回路と、前記数のガイドテーブルエントリを含むガイドテーブルであって、それぞれのガイドテーブルエントリがアドレス値を含み、前記テーブルポインタが最初に前記ガイドテーブル内の第1ガイドテーブルエントリをポイントする、前記ガイドテーブルと、前記制御回路と、前記ガイドテーブルと、に結合せしめられたアドレス発生回路であって、該アドレス発生回路が、前記パケット転送パラメータから、前記スタートアドレスと、前記ディメンション値と、前記数のガイドテーブルエントリと、前記テーブルポインタとを受

ける動作をし、前記アドレス発生回路が、前記ガイドテーブルエントリに対応するメモリアクセス用のアドレスのブロックの集合を、前記スタートアドレスと、前記テーブルポインタによりポイントされた前記ガイドテーブルエントリの前記アドレス値と、の所定の組合せから、ブロックスタートアドレスであって、該第1ブロックスタートアドレスが前記スタートアドレスである、前記ブロックスタートアドレスを形成し、該ブロックスタートアドレスと、前記ディメンション値とからアドレスのブロックを形成し、アドレスの該ブロックにメモリアクセスを行い、もしガイドテーブルエントリの前記数よりも少ないメモリアクセスしか行われ終わっていなければ、前記メモリアクセスに続いて、前記ガイドテーブルの次のエントリをポイントするように前記テーブルポインタを更新する、ことによって形成する、前記アドレス発生回路と、を含む、データ処理装置。

【0456】(16) 前記アドレス発生回路が、前記アドレス値を前記前のブロックスタートアドレスに加算することにより、前記スタートアドレスと、前記テーブルポインタによりポイントされた前記ガイドテーブルエントリの前記アドレス値と、の前記所定の組合せを形成する、第15項記載のデータ処理装置。

【0457】(17) 前記アドレス発生回路が、前記テーブルポインタによりポイントされた前記ガイドテーブル値を前記スタートアドレスに加算することにより、前記スタートアドレスと、前記テーブルポインタによりポイントされた前記ガイドテーブルエントリの前記アドレス値と、の前記所定の組合せを形成する、第15項記載のデータ処理装置。

【0458】(18) 前記パケット転送パラメータの前記ディメンション値が、画素のアレイの水平ディメンション値および垂直ディメンション値を含む、第15項記載のデータ処理装置。

【0459】(19) 前記メモリアクセスが、アドレスの前記ブロックからのメモリアドレスから成る、第15項記載のデータ処理装置。

【0460】(20) 前記メモリアクセスが、アドレスの前記ブロックへのメモリアドレスから成る、第15項記載のデータ処理装置。

【0461】(21) 単一半導体チップ上に、メモリと、パケット転送リクエストを発生する回路を有するデータプロセッサと、前記メモリと、前記データプロセッサと、に接続されたデータ転送制御装置であって、該データ転送制御装置が、パケット転送リクエストと、パケット転送パラメータとを受ける動作をする制御回路であって、該パケット転送パラメータが、スタートアドレスと、アドレスのブロックを定義するディメンション値と、ある数のガイドテーブルエントリと、テーブルポインタとを含む、前記制御回路と、前記数のガイドテーブルエントリを含むガイドテーブルであって、それぞれの

ガイドテーブルエントリがアドレス値を含み、前記テーブルポインタが最初に前記ガイドテーブル内の第1ガイドテーブルエントリをポイントする、前記ガイドテーブルと、前記制御回路と、前記ガイドテーブルと、に結合せしめられたアドレス発生回路であって、該アドレス発生回路が、前記パケット転送パラメータから、前記スタートアドレスと、前記ディメンション値と、前記数のガイドテーブルエントリと、前記テーブルポインタとを受ける動作をし、前記アドレス発生回路が、前記ガイドテーブルエントリに対応するメモリアクセス用のアドレスのブロックの集合を、前記スタートアドレスと、前記テーブルポインタによりポイントされた前記ガイドテーブルエントリの前記アドレス値と、の所定の組合せから、ブロックスタートアドレスであって、該第1ブロックスタートアドレスが前記スタートアドレスである、前記ブロックスタートアドレスを形成し、該ブロックスタートアドレスと、前記パケット転送パラメータの前記ディメンション値とからアドレスのブロックを形成し、アドレスの該ブロックにメモリアクセスを行い、もしガイドテーブルエントリの前記数よりも少ないメモリアクセスしか行われ終わっていなければ、前記メモリアクセスに続いて、前記ガイドテーブルの次のエントリをポイントするように前記テーブルポインタを更新する、ことによって形成する、前記アドレス発生回路と、を含む、前記データ転送制御装置と、を備えた、集積回路。

【0462】(22) 前記アドレス発生回路が、前記アドレス値を前記前のブロックスタートアドレスに加算することにより、前記スタートアドレスと、前記テーブルポインタによりポイントされた前記ガイドテーブルエントリの前記アドレス値と、の前記所定の組合せを形成する、第21項記載の集積回路。

【0463】(23) 前記アドレス発生回路が、前記テーブルポインタによりポイントされた前記ガイドテーブル値を前記スタートアドレスに加算することにより、前記スタートアドレスと、前記テーブルポインタによりポイントされた前記ガイドテーブルエントリの前記アドレス値と、の前記所定の組合せを形成する、第21項記載の集積回路。

【0464】(24) 前記パケット転送パラメータの前記ディメンション値が、画素のアレイの水平ディメンション値および垂直ディメンション値を含む、第21項記載の集積回路。

【0465】(25) 前記メモリアクセスが、アドレスの前記ブロックからのメモリアドレスから成る、第21項記載の集積回路。

【0466】(26) 前記メモリアクセスが、アドレスの前記ブロックへのメモリアドレスから成る、第21項記載の集積回路。

【0467】(27) 前記単一半導体チップ上に含まれない外部メモリにアクセスする動作をする外部メモリイ

ンタフェースをさらに含み、前記アドレス発生回路が、前記メモリおよび前記外部メモリの双方を含むアドレスを発生する、第21項記載の集積回路。

【0468】(28)複数の位置に情報を記憶するステップと、パケットリクエストを受けて、スタートアドレスと、制御信号と、パッチ値と、を供給するステップと、該スタートアドレスと、該制御信号と、該パッチ値と、にตอบสนองしてパッチアドレスを発生するステップと、を含む、画像処理装置を操作する方法。

【0469】(29)本発明は、メモリアクセスのアドレスの制御様式に関する。本発明のデータ処理装置は、メモリと、制御回路と、ガイドテーブルと、アドレス発生回路と、を含む。該制御回路は、パケット転送リクエストと、パケット転送パラメータとを受ける。該パケット転送パラメータは、スタートアドレスと、ある数のガイドテーブルエントリと、テーブルポインタとを含む。前記ガイドテーブルは、ガイドテーブルエントリを含み、それぞれのガイドテーブルエントリは、アドレス値およびアドレスのブロックを定義するディメンション値を含む。前記テーブルポインタは、最初に前記ガイドテーブル内の第1ガイドテーブルエントリをポイントする。前記アドレス発生回路は、前記スタートアドレスと、前記ガイドテーブルエントリの前記アドレス値と、の所定の組合せから形成されるスタートアドレスを有する、それぞれのガイドテーブルエントリに対応するメモリアクセス用のアドレスのブロックの集合を形成する。アドレスの該ブロックは、前記ディメンション値から形成される。前記メモリアクセスに続いて、前記アドレス発生回路は、前記ガイドテーブルの次のエントリをポイントするように前記テーブルポインタを更新する。前記アドレス発生回路は、随意選択的に、前記アドレス値を前記前のブロックスタートアドレスに加算することにより、または、前記ガイドテーブル値を前記スタートアドレスに加算することにより、スタートアドレスと、ガイドテーブルエントリのアドレス値と、の前記所定の組合せを形成しうる。前記メモリアクセスは、アドレスの前記ブロックからのメモリリード、または、アドレスの前記ブロックへのメモリライトでありうる。実施例においては、メモリと、データプロセッサと、上述のメモリアクセスを行うデータ転送制御装置とは、単一半導体チップ内に構成される。該データ転送制御装置は、オンチップメモリと同様に、外部メモリにアクセスしうる。

【0470】警告：著作権1991年テキサスインスツルメンツ社 本特許文献の開示の一部は、著作権およびマスクワーク保護の対象となる資料を含む。この著作権およびマスクワークの所有者は、米国特許庁の特許ファイルまたは記録にある特許文献または特許の開示の、何人によるコピーに反対するものではないが、それ以外に対しては、すべての著作権およびマスクワークの権利を留保するものである。

【0471】関連出願とのクロスレファレンス：本願は、下記の米国特許および継続中の米国特許出願に開示された発明の改良に関するもので、下記の特許および米国特許出願はすべてテキサスインスツルメンツ社に譲渡されたものであり、これらのいずれも参考例として援用する。

【0472】1994年6月21日出願された「プロセッサとメモリをクロスパーリンクしたマルチプロセッサおよびその作動方法」を発明の名称とする、米国特許出願第08/263,501号。この出願は、1993年10月12日出願され現在放棄された米国特許出願第08/135,754号の継続出願であり、1992年8月21日出願され現在放棄されている米国特許出願第07/933,865号の継続出願であり、この米国特許出願第07/933,865号は、1989年11月17日出願され現在放棄されている米国特許出願第07/435,591号の継続出願である。

【0473】1989年11月17日出願され1993年5月18日に発行された「SIMD/MIMD再構成可能なマルチプロセッサおよびオペレーション方法」を発明の名称とする米国特許第5,212,777号。

【0474】1989年11月17日出願され現在放棄されている米国特許出願第07/437,856号の継続出願である、1992年6月5日出願され現在放棄されている米国特許出願第07/895,565号の継続出願である、1994年6月22日出願された「マルチプロセッサ用再構成可能な通信およびそのオペレーション方法」を発明の名称とする米国特許出願第08/264,111号。

【0475】1989年11月17日出願され現在放棄されている米国特許出願第07/437,852号の継続出願である、1994年6月22日出願された「小エリアのクロスパーおよびそのオペレーション方法」を発明の名称とする米国特許出願第08/264,582号。

【0476】1989年11月17日出願され現在放棄されている米国特許出願第07/437,853号の継続出願である、1993年5月15日出願された「同期されたMIMDマルチ処理システムおよびそのオペレーション方法」を発明の名称とする米国特許出願第08/032,530号。

【0477】1989年11月17日出願され、1993年3月23日発行された「スライスされたアドレス指定用マルチプロセッサおよびそのオペレーション方法」を発明の名称とする米国特許第5,197,140号。

【0478】1989年11月17日出願され、1994年8月16日発行された「画像データの二進ストリーム内の1の数をカウントするための相互接続された半加算器のマトリックスを利用する1をカウントする回路」を発明の名称とする米国特許第5,339,447号。

【0479】1989年11月17日出願され、1993年8月24日に発行された「SIMDモードで作動する際にデータメモリとしてMIMDインストラクションメモリを再利用するデュアルモードのSIMD/MIMDプロセッサ」を發明の名称とする米国特許第5,239,654号。

【0480】1989年11月17日出願され現在放棄されている米国特許出願第437,854号の継続出願である、1992年6月29日に出版された「イメージングコンピュータおよびそのオペレーション方法」を發明の名称とする米国特許出願第07/911,562号。

【0481】1989年11月17日出願され、1993年7月6日に発行された「集積クロスポイントロジックを有するスイッチマトリックスおよびそのオペレーション方法」を發明の名称とする米国特許第5,226,125号。

【0482】1993年11月30日出願された「パレルローデータに備えた3入力端論理ユニット」を發明の名称とする米国特許出願第08/160,299号。

【0483】1993年11月30日出願された「複数の独立部分および各部分からの結果の表示ビットを記憶するレジスタを有する算術論理ユニット」を發明の名称とする米国特許出願第08/158,742号。

【0484】1993年11月30日出願された「レジスタペア条件からのメモリストア」を發明の名称とする米国特許出願第08/160,118号。

【0485】1993年11月30日出願され現在放棄されている米国特許出願第08/160,115号の継続出願である、1994年10月17日出願された「繰返しごと複数の商ビットを形成する繰返し割り算装置、システムおよび方法」。

【0486】1993年11月30日出願された「混合された算術およびブール組み合わせを形成する3入力端算術論理ユニット」を發明の名称とする米国特許出願第08/159,285号。

【0487】1993年11月30日出願された「単一データワードの複数の等しい部分におけるデータの合計を計算する方法、装置およびシステム」を發明の名称とする米国特許出願第08/160,119号。

【0488】1993年11月30日出願された「サイズ検出のため最小位ビットの変化を用いるハフマンコード化方法、回路およびシステム」を發明の名称とする米国特許出願第08/159,359号。

【0489】1993年11月30日出願された「負の数を変換するため条件付き減算を利用するハフマンコード化方法、回路およびシステム」を發明の名称とする米国特許出願第08/160,296号。

【0490】1993年11月30日出願された「複数の絶対値の差を合計するための方法、装置およびシステ

ム」を發明の名称とする米国特許出願第08/160,112号。

【0491】1993年11月30日出願された「排他的ORによる最も左側の1の検出を利用する繰返し割り算装置、システムおよび方法」を發明の名称とする米国特許出願第08/160,120号。

【0492】1993年11月30日出願された「2つの独立アドレスの選択的マージを利用するアドレス発生器」を發明の名称とする米国特許出願第08/160,114号。

【0493】1993年11月30日出願された「相関化方法、装置およびシステム」を發明の名称とする米国特許出願第08/160,116号。

【0494】1993年11月30日出願された「複数の独立したプロセッサオペレーションを制御する長いインストラクションワード」を發明の名称とする米国特許出願第08/160,297号。

【0495】1993年11月30日出願された「直交データ変換のための回転レジスタ」を發明の名称とする米国特許出願第08/159,346号。

【0496】1993年11月30日出願された「メディアアンフィルタ方法、回路およびシステム」を發明の名称とする米国特許出願第08/159,652号。

【0497】1993年11月30日出願された「条件レジスタソース選択を備えた算術論理ユニット」を發明の名称とする米国特許出願第08/159,344号。

【0498】1993年11月30日出願された「繰返しによる除算のための装置、システムおよび方法」を發明の名称とする米国特許出願第08/160,301号。

【0499】1993年11月30日出願された「冗長コード化された乗算の結果を利用する乗算丸め方法」を發明の名称とする米国特許出願第08/159,650号。

【0450】1993年11月30日出願された「スプリット乗算方法」を發明の名称とする米国特許出願第08/159,349号。

【0451】1993年11月30日出願された「ゼロの条件テストを含む混合形条件テストおよびブランチ演算」を發明の名称とする米国特許出願第08/158,741号。

【0452】1993年11月30日出願された「パックされたワード対の乗算方法」を發明の名称とする米国特許出願第08/160,302号。

【0453】1993年11月30日出願された「シフタを備えた3入力端算術論理ユニット」を發明の名称とする米国特許出願第08/160,573号。

【0454】1993年11月30日出願された「マスク発生器を備えた3入力端算術論理ユニット」を發明の名称とする米国特許出願第08/159,282号。

【0455】1993年11月30日出願された「バレルローテータおよびマスク発生器を備えた3入力端算術論理ユニット」を發明の名称とする米国特許出願第08/160,111号。

【0456】1993年11月30日出願された「シフトおよびマスク発生器を備えた3入力端算術論理ユニット」を發明の名称とする米国特許出願第08/160,298号。

【0457】1993年11月30日出願された「第2入力と第3入力のブール組み合わせプラス第2入力と第3入力の第2ブール組み合わせが加算された第1入力の合計を計算する3入力端算術論理ユニット」を發明の名称とする米国特許出願第08/159,345号。

【0458】1993年11月30日出願された「第1、第2および第3入力のブール組み合わせプラス第1、第2、第3入力の第2ブール組み合わせの合計を計算する3入力端算術論理ユニット」を發明の名称とする米国特許出願第08/160,113号。

【0459】1993年11月30日出願された「桁上げ伝搬ロジックを利用する3入力端算術論理ユニット」を發明の名称とする米国特許出願第08/159,640号。

【0460】1993年11月30日出願された「書き込み優先権を利用するIF、THENオペレーションのためのデータ処理装置、システムおよび方法」を發明の名称とする米国特許出願第08/160,300号。

【0461】1993年3月8日出願された「MPベクトルインストラクションFP+LOAD/STORE」を發明の名称とする米国特許出願第08/207,989号(TI整理番号15521)。

【0462】1993年3月8日出願された「不動点少数のための正規化方法」を發明の名称とする米国特許出願第08/ 号(TI整理番号18695)。

【0463】本願は下記の米国特許出願とも関連しており、これらのいずれも参考例としてここに援用する。

【0464】1993年3月8日出願された「PP転送プロセッサにおけるトランスペアレンシーおよび平面マスキング」を發明の名称とする米国特許出願第08/208,413号。

【0465】1993年3月8日出願された「トランスペアレンシーを備えたPIXBLT」を發明の名称とする米国特許出願第08/208,161号。

【0466】1993年3月8日出願された「プロセッサからのメッセージパッシングおよびブラストインタラプト」を發明の名称とする米国特許出願第08/208,171号。

【0467】1993年3月8日出願された「X、Y次元を備えたガイド転送および変数ステップング」を發明の名称とする米国特許出願第08/209,123号。

【0468】1993年3月8日出願された「ガイド転

送ラインドロ잉」を發明の名称とする米国特許出願第08/209,124号。

【0469】1993年3月8日出願された「異なるメモリタイプを同時に制御する転送プロセッサ用メモリインターフェース」を發明の名称とする米国特許出願第08/208,517号。

【0470】1993年3月8日出願された「TP転送プロセッサのアーキテクチャ」を發明の名称とする米国特許出願第08/207,503号。

【図面の簡単な説明】

【図1】本發明の画像処理システムのシステムアーキテクチャを示す図。

【図2】単一集積回路マルチプロセッサのアーキテクチャを示す図。

【図3】デジタル画像/グラフィックプロセッサのインタラプトイネーブルおよびインタラプトフラグレジスタを示す図。

【図4】共通ワードのフィールドを示す図。

【図5】デジタル画像/グラフィックプロセッサの通信レジスタを示す図。

【図6】画像システムバスの優先度およびオペレーション方法を示す図。

【図7】クロスバー優先度およびマルチプロセッサ集積回路内のオペレーション方法を示す図。

【図8】本發明の実施例の転送プロセッサ構造を示す図。

【図9】LASTPAGEレジスタの使用を示す図。

【図10】外部メモリシーケンサのステート図およびオペレーション方法を示す図。

【図11】ソースマシン構造を示す図。

【図12】パケット転送FIFOバッファ、キャッシュバッファ算術演算およびブロックの例を示す図。

【図13】バッファカウンタ算術演算およびブロックの例を示す図。

【図14】aは小エンディアンパケット転送FIFOバッファの例および方法を示す図。bは大エンディアンパケット転送FIFOバッファの例および方法を示す図。

【図15】プロセッサのパラメータメモリの使用例を示す図。

【図16】リンクされたリストパケット転送方法の一例を示す図。

【図17】次元の定められたパケット転送方法の一例を示す図。

【図18】固定パッチのデルタガイドパケット転送方法の一例を示す図。

【図19】固定パッチのオフセットガイド転送方法の一例を示す図。

【図20】固定パッチのオフセットガイドルックアップテーブルパケット転送方法の一例を示す図。

【図21】小エンディアン可変パッチガイドテーブルフ

フォーマットを示す図。

【図22】大エンディアン可変パッチガイドテーブルフォーマットを示す図。

【図23】可変パッチデルタガイドパケット転送方法の一例を示す図。

【図24】可変パッチオフセットガイドパッチ転送方法の一例を示す図。

【図25】ソーストランスペアレンシーパラメータを備えた、次元の定められたパケット転送の一例を示す図。

【図26】フィルパラメータを備えた、次元の定められたパケット転送の一例を示す図。

【図27】次元の定められたソースおよび固定パッチガイド宛て先パケット転送パラメータの一例を示す図。

【図28】次元の定められたソースおよび可変パッチガイド宛て先パケット転送パラメータの一例を示す図。

【図29】次元の定められたブロック書き込みパケット転送パラメータの一例を示す図。

【図30】パケット転送パラメータのエンコーディングを示す図。

【図31】パケット転送タイマー構造を示す図。

【図32】一時中断されたパケット転送パラメータの記憶方法の一例を示す図。

【図33】一時中断されたパケットサービスパラメータのエンコーディングを示す図。

【図34】小エンディアンモードでの8×ブロック書き*

* 込みビット再マッピングにおけるビット再マッピングの一例を示す図。

【図35】小エンディアンモードでの4×ブロック書き込みビット再マッピングにおけるビット再マッピングの一例を示す図。

【図36】シリアルレジスタ転送方法の一例を示す図。

【図37】トランスペアレンシーオペレーションを示す図。

【図38】トランスペアレンシーおよびバイト書き込みロジック回路の実施例を示す図。

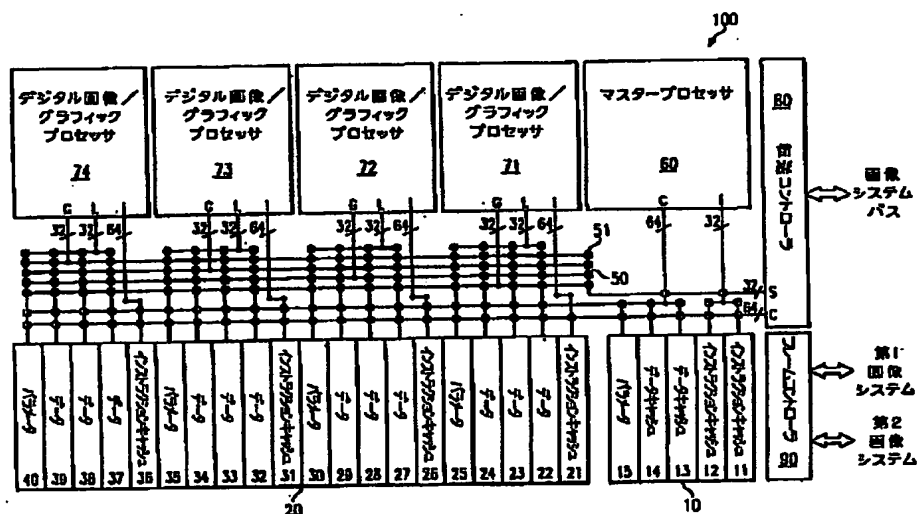
【図39】高品位テレビシステムの一実施例を示す図。

【図40】単一のデジタル画像/グラフィックプロセッサを有するマルチプロセッサ集積回路を含むカラーファクシミリシステムの一実施例を示す図。

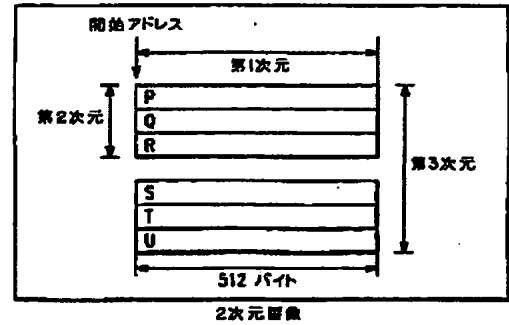
【符号の説明】

- 1 ホスト処理システム
- 2 ホスト周辺デバイス
- 3 撮像デバイス
- 4 画像キャプチャコントローラ
- 5、6 ビデオラム
- 7 ビデオパレット
- 8 ビデオディスプレイ
- 9 メモリ
- 16 トランシーバ
- 100 マルチプロセッサ集積回路

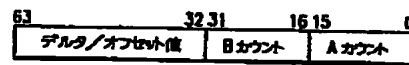
【図2】



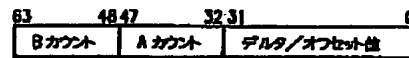
【圖 17】



【図 2 1】



【图 2 2】

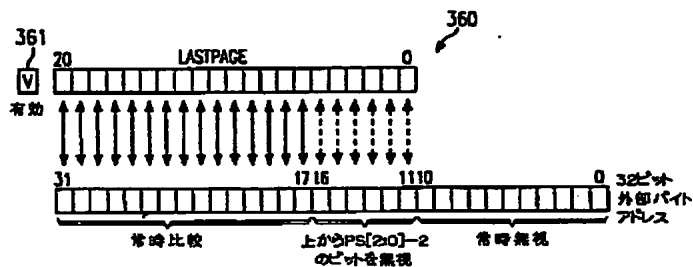


31 28 120 15 8 2 1 0

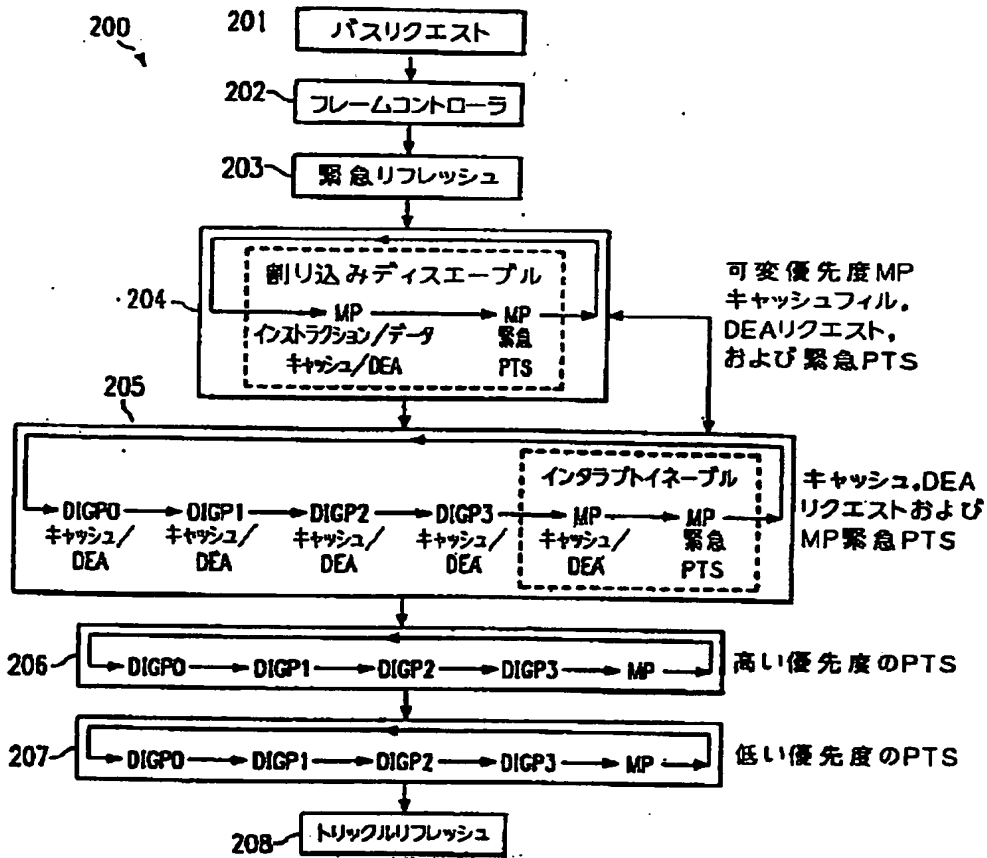
F S Q P - - - - - 同期ビット - - - - - r DIGN

⎵ パケット転送 r r r r 0 0 0 0 DIGN
⎵ PT, 待機 c c c c 1 1 1 1 番号
⎵ 一時中断 s s s s G G G G
⎵ フォアグラウンド J 2 1 0 P P P P

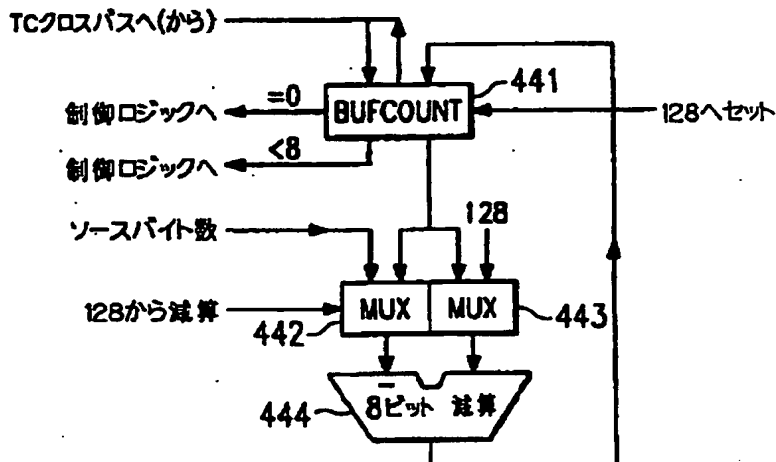
【図9】



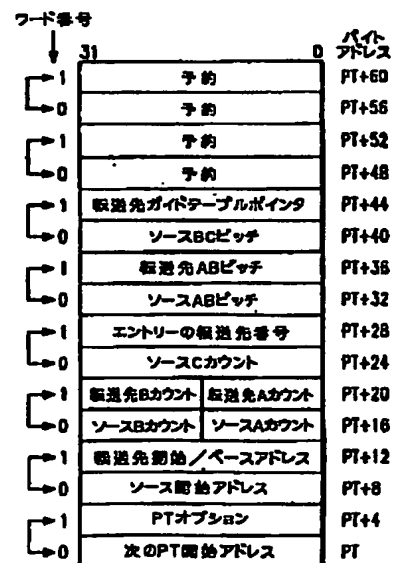
【図6】



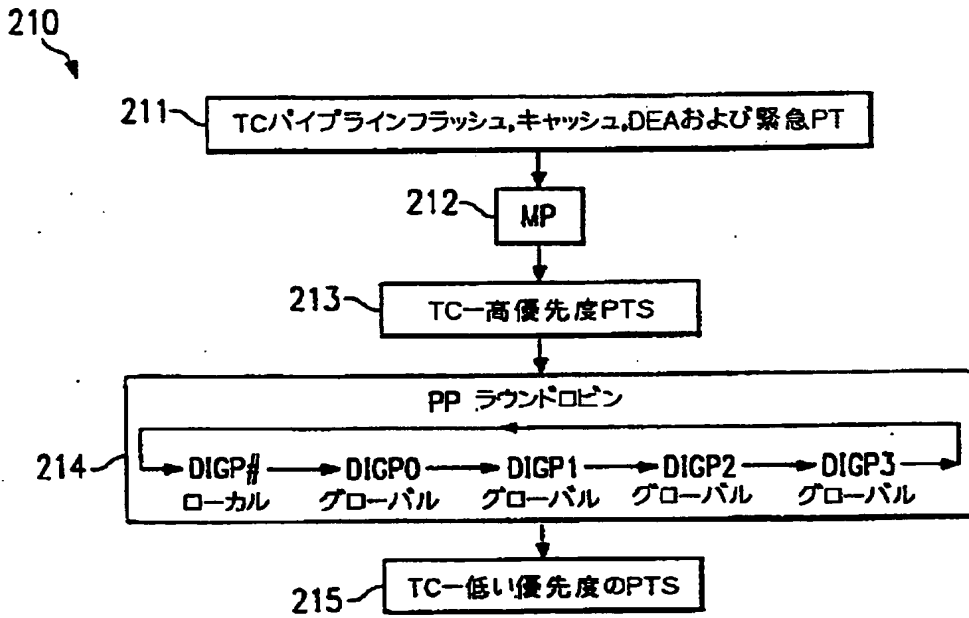
【図13】



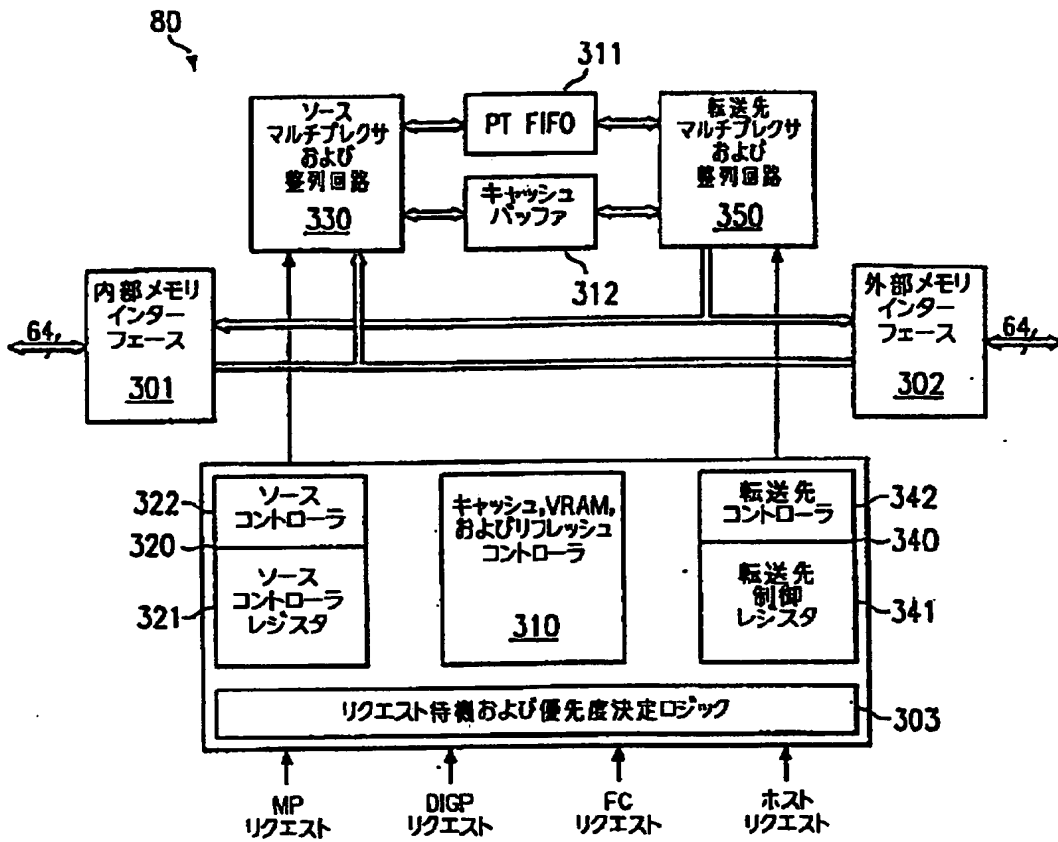
【図27】



【図7】



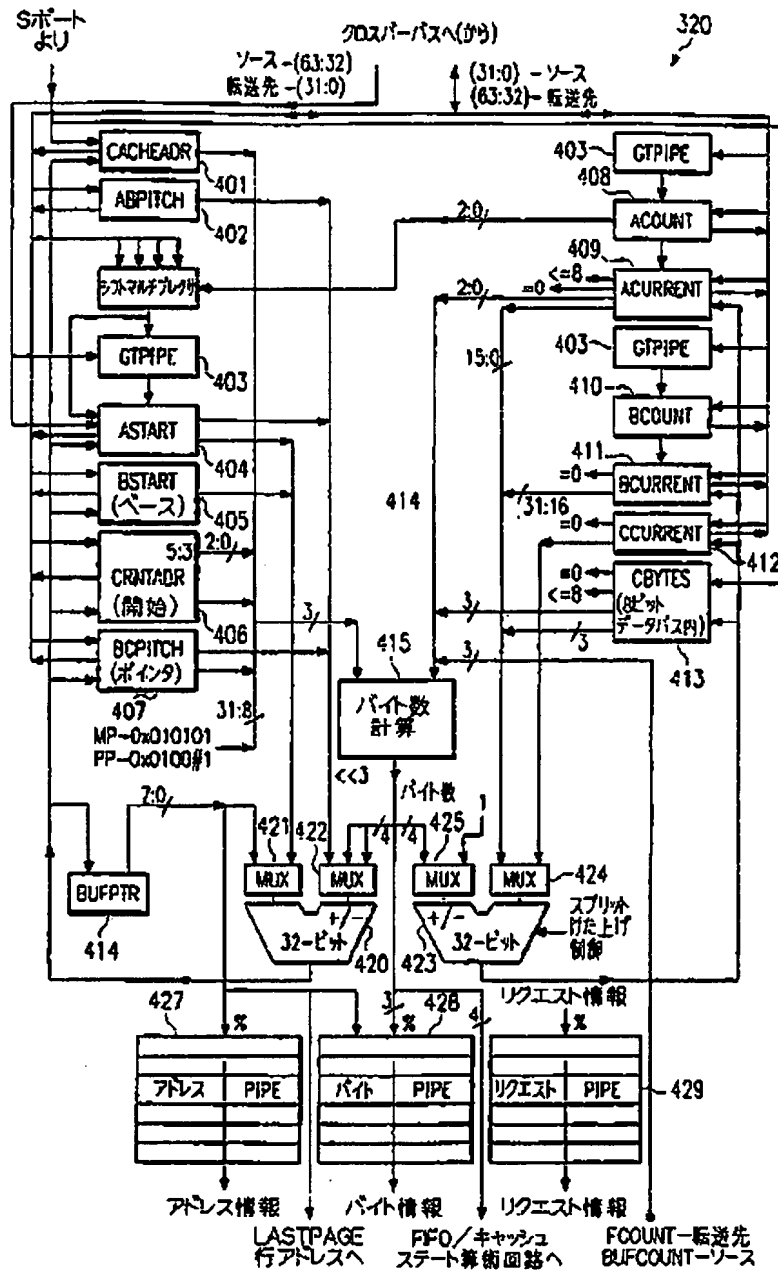
【図8】



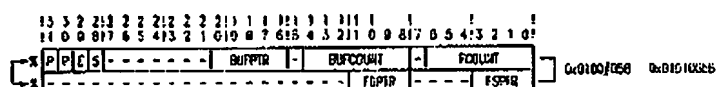
(80)

特開平8-77347

【図11】



【図33】



[illegible]

```

graph LR
    A[リスト開始ポインタ] --> B[第1パケットパラメータ  
テーブル転送]
    B --> C[最終パケットパラメータ  
テーブル転送]
    C --> D[16進数 "XXXXXXXX"]
    subgraph RAM [パラメータRAM]
        A
    end
    subgraph Memory [仕方のオンチップメモリ]
        B
        C
        D
    end
  
```

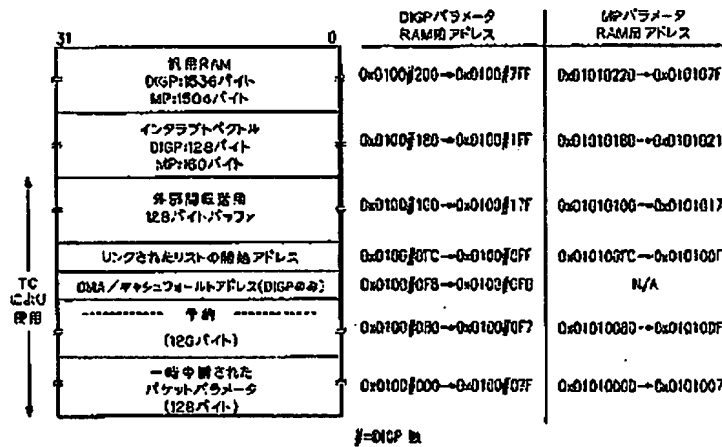
ワード番号		バイト アドレス
↓	31	0
1	予約	PT+50
0	予約	PT+56
1	トランスベレーションワード1	PT+52
0	トランスベレーションワード0	PT+48
1	転送先Bピッチ	PT+44
0	ソースBピッチ	PT+40
1	転送先ABピッチ	PT+36
0	ソースABピッチ	PT+32
1	転送先Cカウンタ	PT+28
0	ソースCカウンタ	PT+24
1	転送先Bカウンタ	PT+20
0	ソースBカウンタ	PT+16
1	転送先開始アドレス	PT+12
0	ソース開始アドレス	PT+8
1	PTオプション	PT+4
0	次のPT開始アドレス	PT

ワード1	ワード0	小エンディアン
ワード0	ワード1	大エンディアン

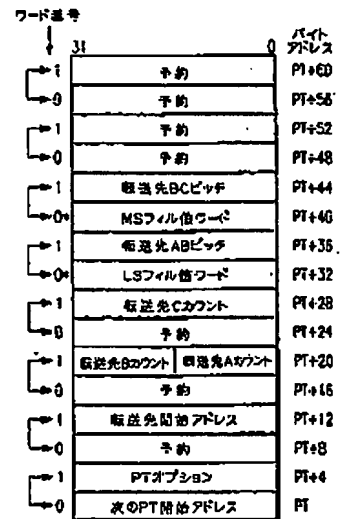
(82)

特開平8-77347

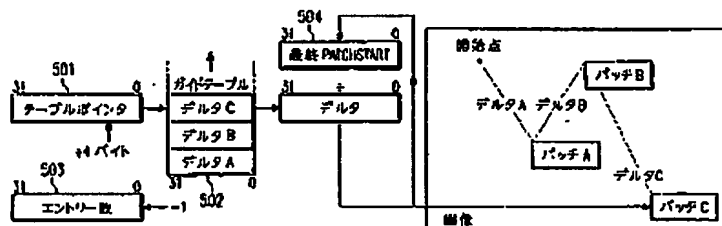
【図15】



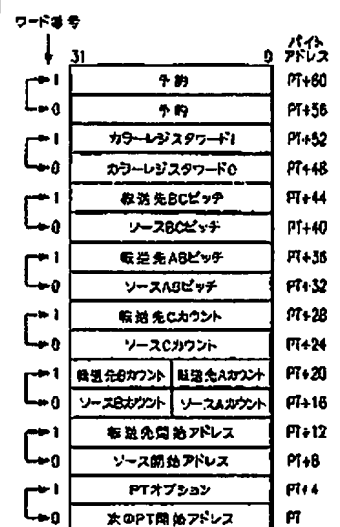
【図26】



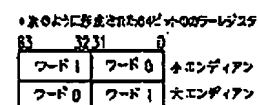
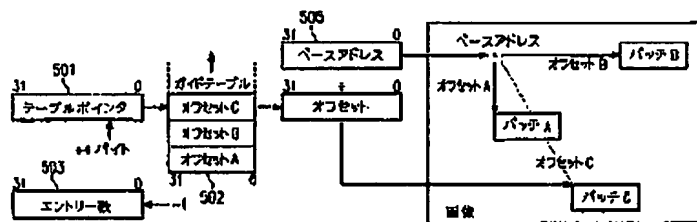
【図18】



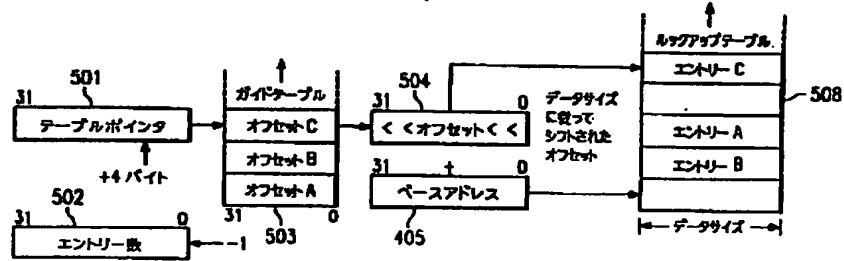
【図29】



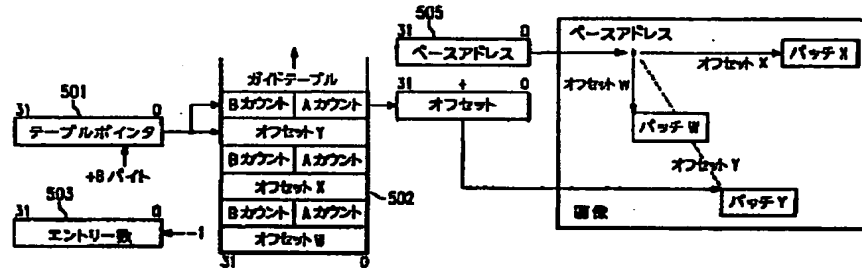
【図19】



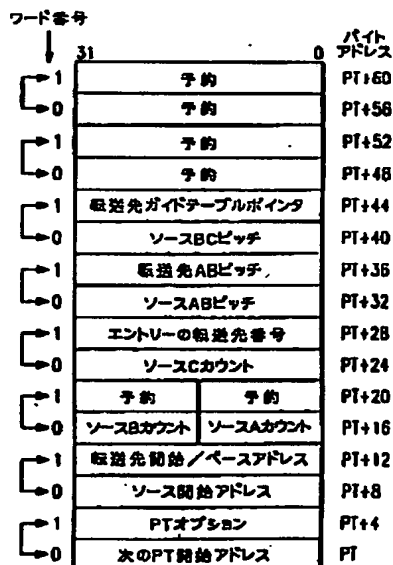
【図20】



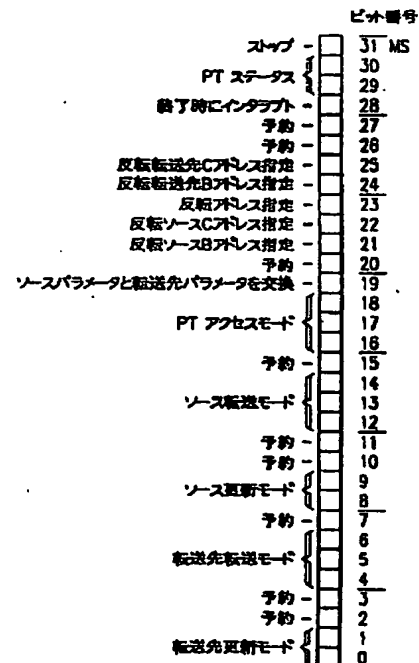
【図24】



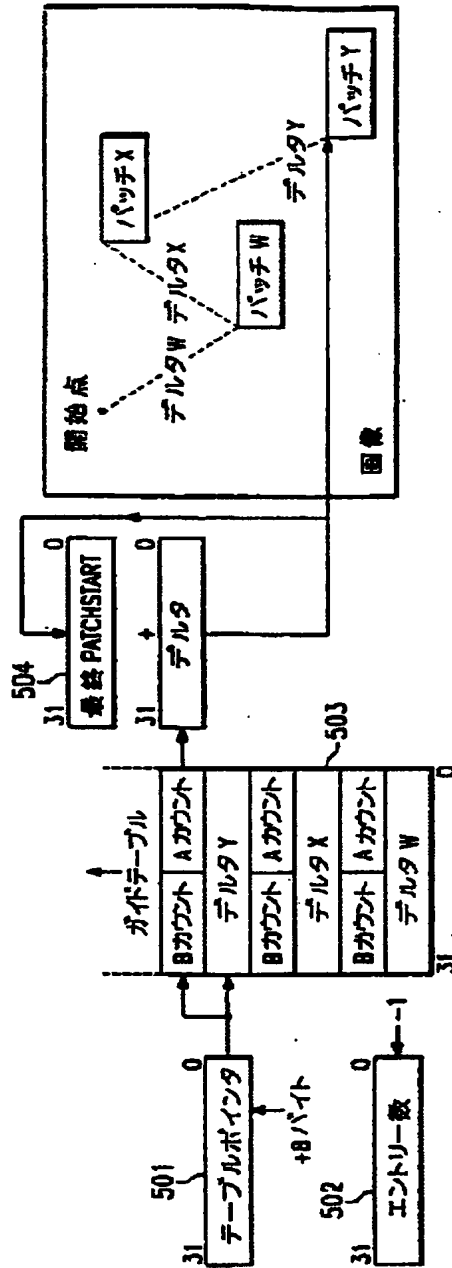
【図28】



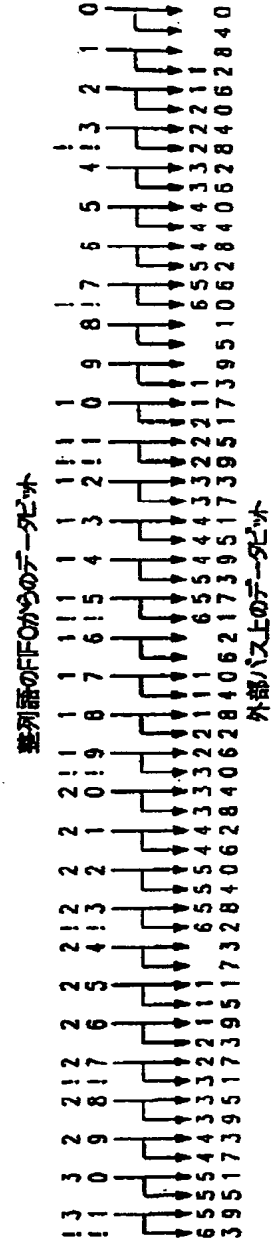
【図30】



【図23】



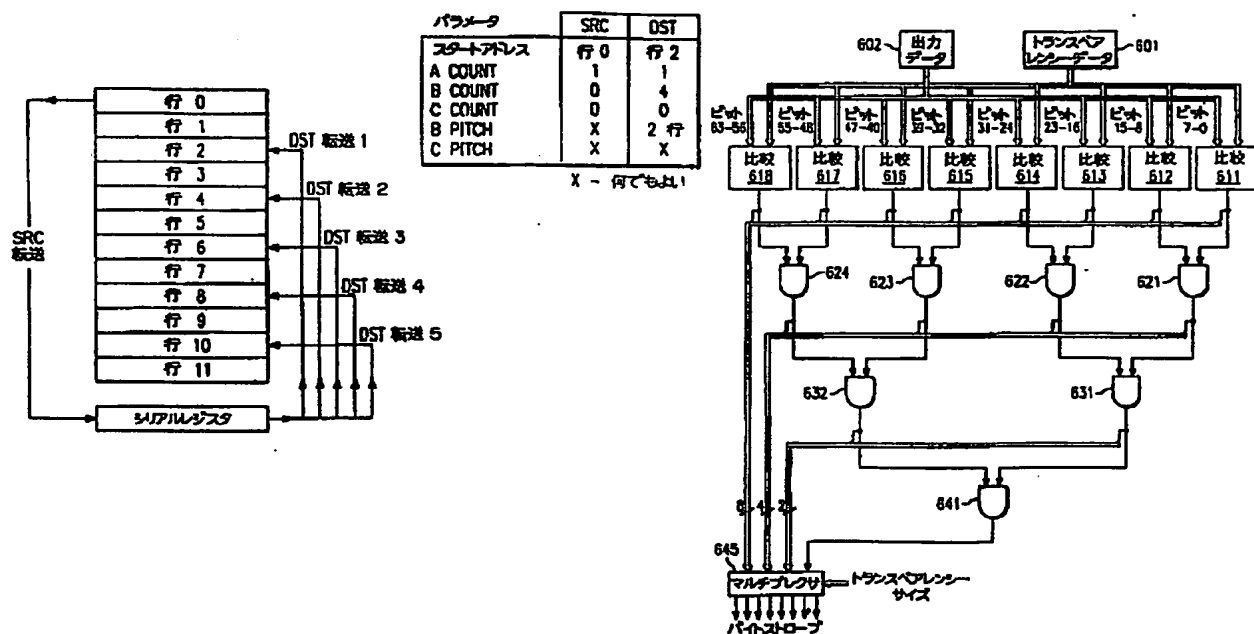
【図35】



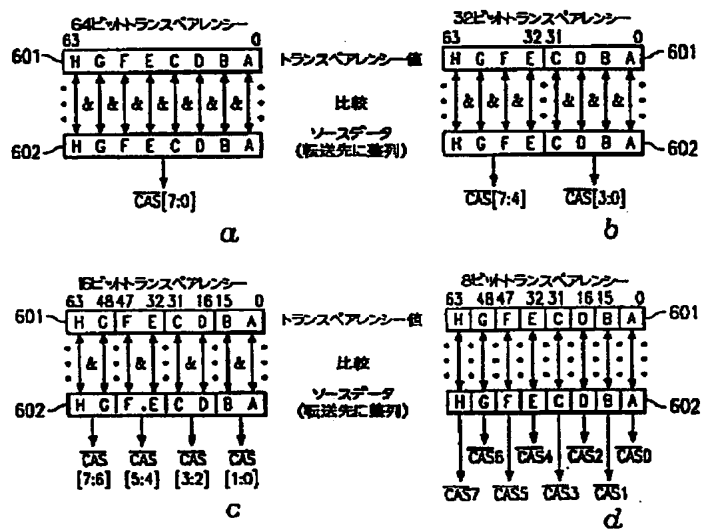
【图 3 2】



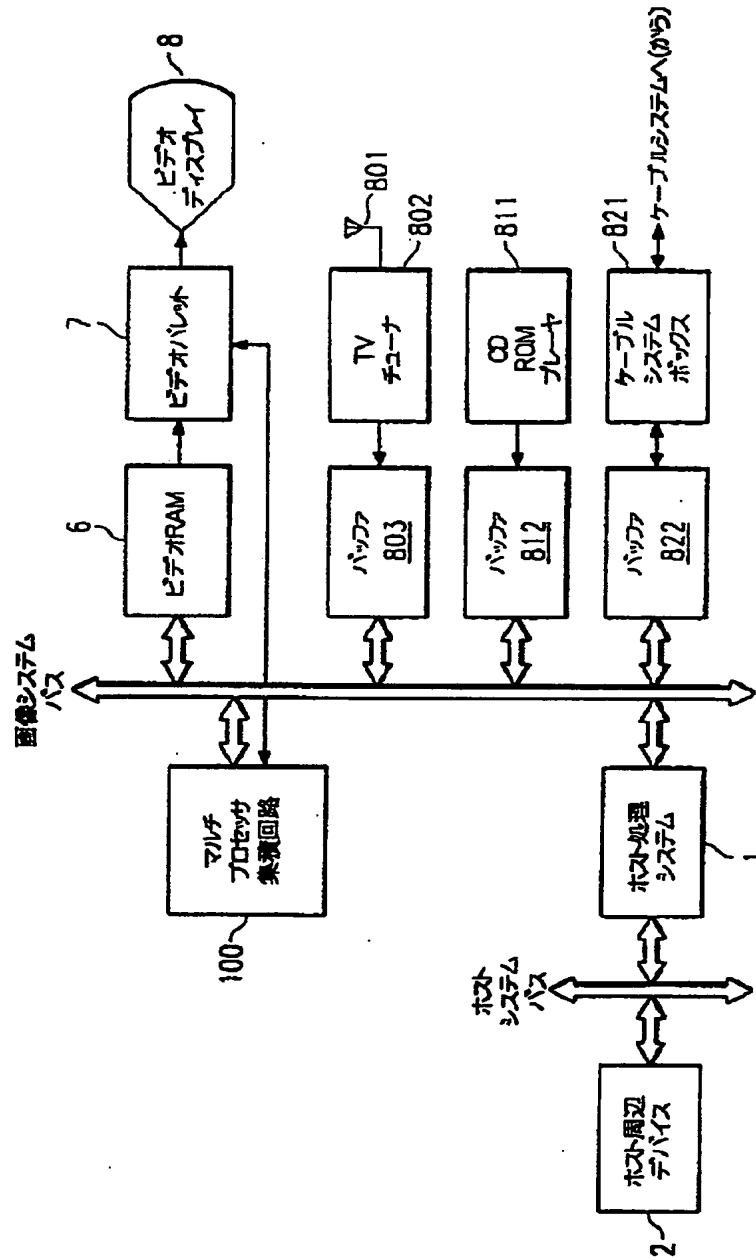
【图 38】



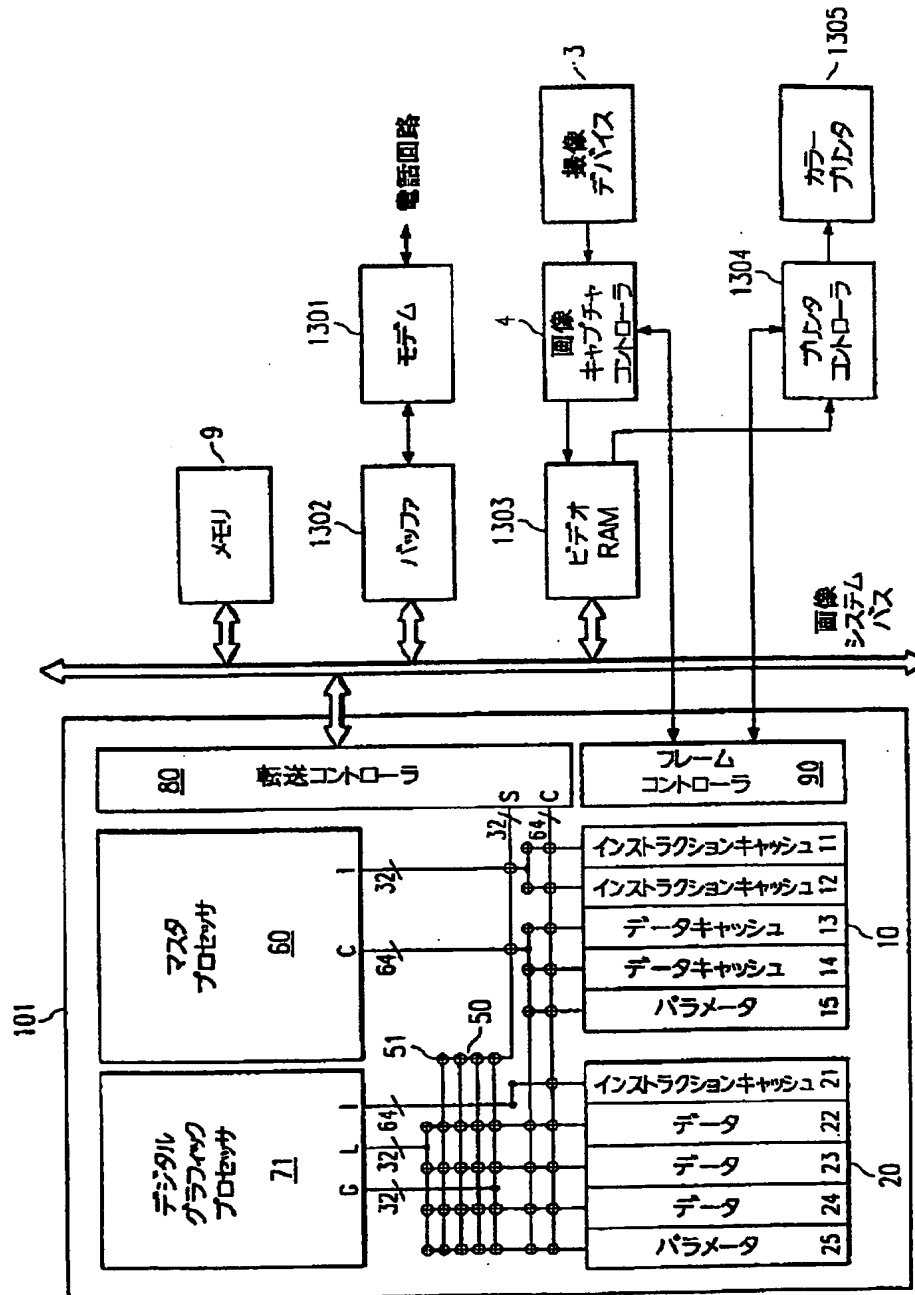
【図 37】



【図39】



【図40】



【手続補正書】

【提出日】平成7年6月8日

【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】全文

【補正方法】変更

【補正内容】

【書類名】明細書

【発明の名称】画像／グラフィックス処理用のデータ処理装置およびその操作方法

【特許請求の範囲】

【請求項1】複数の対応するアドレスにデータを記憶するメモリと、

パケット転送リクエストと、パケット転送パラメータとを受ける動作をする制御回路であって、該パケット転送パラメータが、スタートアドレスと、ある数のガイドテーブルエン트리と、テーブルポインタとを含む、前記制御回路と、

前記数のガイドテーブルエントリを含むガイドテーブルであって、それぞれのガイドテーブルエントリが、アドレス値およびアドレスのブロックを定義するディメンション値を含み、前記テーブルポインタが最初に前記ガイドテーブル内の第1ガイドテーブルエントリをポイントする、前記ガイドテーブルと、

前記制御回路に結合せしめられたアドレス発生回路であって、該アドレス発生回路が、前記パケット転送パラメータから、前記スタートアドレスと、前記数のガイドテーブルエントリと、前記テーブルポインタとを受ける動作をし、前記アドレス発生回路が、前記ガイドテーブルエントリに対応するメモリアクセス用のアドレスのブロックの集合を、

前記スタートアドレスと、前記テーブルポインタによりポイントされた前記ガイドテーブルエントリの前記アドレス値と、の所定の組合せから、ブロックスタートアドレスであって、該第1ブロックスタートアドレスが前記スタートアドレスである、前記ブロックスタートアドレスを形成し、

該ブロックスタートアドレスと、前記テーブルポインタによりポイントされた前記ガイドテーブルエントリの前記ディメンション値と、からアドレスのブロックを形成し、

アドレスの該ブロックにメモリアクセスを行い、もしガイドテーブルエントリの前記数よりも少ないメモリアクセスしか行われ終わっていなければ、前記メモリアクセスに続いて、前記ガイドテーブルの次のエントリをポイントするように前記テーブルポインタを更新することによって形成する、前記アドレス発生回路と、を含む、データ処理装置。

【請求項2】 複数の位置に情報を記憶するステップと、

パケットリクエストを受けてスタートアドレスおよび制御信号を供給するステップと、

該スタートアドレスと、制御信号と、パッチ値と、にตอบสนองしてパッチアドレスを発生するステップと、

パッチ情報をデコードしてディメンション値およびパッチ値を供給するステップと、

前記スタートアドレスと、前記ディメンション値と、前記パッチ値とにตอบสนองしてパッチアドレスを発生するステップと、を含む、データ処理装置を操作する方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、デジタルデータ処理の技術分野にあり、より詳細には特にデジタル画像／グラフィック処理のようなデジタルデータ処理を行うマイクロプロセッサ回路アーキテクチャおよび方法に関する。

【0002】

【従来技術】本発明の実施例は用途が多く、そのうちのいくつかは説明のための背景として本明細書に説明したコンピュータグラフィックの分野に関する。ビットマップグラフィックとして知られているコンピュータグラフィックの分野では、コンピュータメモリは、画像のうちの個々の画素すなわちピクセルのデータを画像内のそのピクセルの位置に対応するメモリ位置に記憶するようになっている。この画像はディスプレイすべき画像、または操作し、記憶し、ディスプレイし、または再送信する取り込み画像とすることができる。ビットマップコンピュータグラフィックの技術分野は、ダイナミックランダムアクセスメモリ (DRAM) のコストが低下し、かつ記憶容量が増加したこと、更にマイクロプロセッサのコストが低下し、処理パワーが増大したことにより、大幅に有利となった。これらの部品のコストおよび性能が有利に変わったことにより、より大規模で、かつより複雑なコンピュータ画像システムを経済的に実現することが可能となった。

【0003】

【発明が解決しようとする課題】ビットマップグラフィックの分野は、画像データ操作に使用される処理タイプの変遷において、数回の段階を経験している。まず最初に、ビットマップグラフィックをサポートするコンピュータシステムは、すべてのビットマップ演算のためにシステムプロセッサを使用していた。このタイプのシステムには数種の欠点がある。第1の欠点として、コンピュータシステムのプロセッサは、ビットマップグラフィックの取り扱いのために特に設計されたものではないことが挙げられる。汎用計算に極めて妥当なデザイン選択は、ビットマップグラフィックシステムには不適當である。従って、ある種のルーチンググラフィックのタスクは低速でしか作動しない。更にビットマップグラフィックの画像操作のために必要な処理は、他の演算も低速にするほど、システムのプロセッサの計算容量に負荷をかけることが、すぐに判った。

【0004】ビットマップグラフィック処理の革新における次の段階は、専用ハードウェアのグラフィックコントローラにあった。これらデバイスは、システムプロセッサの制御により、簡単な図形、例えばライン、楕円および円を描画できる。これらデバイスのうちで、ピクセルブロック転送 (Pixel) を行うこともできるデバイスもある。ピクセルブロック転送とは、メモリのある部分から他の部分へ画像データをメモリ上で移動する操作のことである。ピクセルブロック転送は、ディスプレイされていないメモリからビットマップディスプレイメモリに転送することにより、標準的画素、例えば特定タイプのフォントのアルファニューメリックキャラクタ

をディスプレイ内に入れるのに有効である。最も頻繁に使用されるグラフィック機能のうちのいくつかを行うためのビルトインアルゴリズムは、システム性能を改善する一つの方法を提供するものである。グラフィックコンピュータシステムは、かかるハードウェアのグラフィックコントローラでほとんど実現されていないような他の機能を有利に含むこともできる。これらの他の機能は、システムプロセッサによりソフトウェアで実現できる。これらのハードウェアグラフィックコントローラでは、一般にシステムプロセッサのビットマップメモリへのごく限られたアクセスを可能とし、これによりシステムソフトウェアがハードウェアのグラフィックコントローラの機能の固定された組の機能を拡張できる程度が制限されている。

【0005】グラフィックシステムプロセッサは、ビットマップグラフィック処理の革新において、別の進歩に寄与した。グラフィックシステムプロセッサとはマイクロプロセッサの属性のすべてを有し、更にビットマップグラフィックのための特殊機能を含むプログラマブルデバイスである。テキサスインスツルメンツ社によって製造されたTMS34010およびTMS34020グラフィックシステムプロセッサは、このクラスのデバイスの代表例である。これらグラフィックシステムプロセッサは、マイクロプロセッサと同様に、記憶されたプログラムに応答するもので、演算論理ユニット、レジスタファイル内のデータ記憶装置ならびにプログラムフローおよび外部データメモリの制御により、データを操作できる能力を有する。更にこれらデバイスは適当なプログラム制御で作動する特殊グラフィックオペレーション用ハードウェアを含む。これらシステムプロセッサのインストラクションセット内の付加インストラクションが、この特殊グラフィック用ハードウェアを制御するようになっている。これらインストラクションおよびこれらをサポートするハードウェアは多くのコンテキストで有効なベースレベルのグラフィック機能を実行するように選択されている。従って特定の問題のために選択されたアルゴリズムを用いることにより、多くの異なるグラフィックアプリケーションに対してプログラムできる。これにより、ハードウェア用コントローラからプログラムマイクロプロセッサに変えることによって得られる有効性の増大と同様に有効性が増す。かかるグラフィックシステムプロセッサは、マイクロプロセッサと同様にプログラム可能なデバイスであるので、これらはスタンドアロングラフィックプロセッサ、システムプロセッサにスレーブ制御されるグラフィックコプロセッサ、または密に結合されたグラフィックコントローラとして作動できる。

【0006】いくつかの分野では、経済的に実現できるように、よりコスト的に効果的であり、パワーのあるグラフィックオペレーションを利用することが好ましい。

これらの例としては、ビデオ会議、フル動画ビデオによるマルチメディア計算、高品位テレビ、カラーファクシミリ、スマートフォトコピー、画像認識システムおよびデジタル写真がある。これらの分野の各々には固有の問題がある。画像データ圧縮およびデコンプレッションの問題はこれらアプリケーションのいくつかにおいて共通のテーマとなっている。画像および特定のフルモーションのビデオに必要な送信バンド幅の値および記憶容量は大きい。許容可能な最終画質を生じさせるような効率的なビデオ圧縮およびデコンプレッションを用いない場合、これらのアプリケーションは送信バンド幅および記憶容量に関連するコストによって制限されてしまう。この分野では、画像処理機能、例えば画像認識およびグラフィック機能、例えばディスプレイ制御の双方をサポートできる単一システムに対するニーズもある。

【0007】

【課題を解決するための手段】本発明は、メモリアクセスのアドレスの制御様式に関する。本発明のデータ処理装置は、メモリと、制御回路と、ガイドテーブルと、アドレス発生回路と、を含む。該制御回路は、パケット転送リクエストと、パケット転送パラメータとを受ける。該パケット転送パラメータは、スタートアドレスと、ある数のガイドテーブルエントリと、テーブルポインタとを含む。前記ガイドテーブルは、ガイドテーブルエントリを含み、それぞれのガイドテーブルエントリは、アドレス値およびアドレスのブロックを定義するディメンション値を含む。前記テーブルポインタは、最初に前記ガイドテーブル内の第1ガイドテーブルエントリをポイントする。前記アドレス発生回路は、それぞれのガイドテーブルエントリに対応するメモリアクセス用のアドレスのブロックの集合を形成する。前記アドレス発生回路は、前記スタートアドレスと、前記テーブルポインタによりポイントされた前記ガイドテーブルエントリの前記アドレス値と、の所定の組合せから、ブロックスタートアドレスを形成する。第1ブロックスタートアドレスは、前記スタートアドレスである。前記アドレス発生回路は、前記ブロックスタートアドレスと、前記テーブルポインタによりポイントされた前記ガイドテーブルエントリの前記ディメンション値と、からアドレスのブロックを形成する。前記アドレス発生回路は、前記ブロックスタートアドレスと、前記ガイドテーブルエントリの前記ディメンション値と、からアドレスのブロックを形成する。前記データ処理装置は、アドレスの該ブロックにメモリアクセスを行う。該メモリアクセスに続いて、前記アドレス発生回路は、前記ガイドテーブルの次のエントリをポイントするように前記テーブルポインタを更新する。

【0008】前記アドレス発生回路は、随意選択的に、前記アドレス値を前記前のブロックスタートアドレスに加算することにより、スタートアドレスと、ガイドテー

ブルエントリのアドレス値と、の前記所定の組合せを形成しうる。これは、デルタガイドドアドレッシングとして公知である。前記アドレス発生回路は、随意選択的に、前記ガイドテーブル値を前記スタートアドレスに加算することにより、スタートアドレスと、ガイドテーブルエントリのアドレス値と、の前記所定の組合せを形成しうる。これは、オフセットガイドドアドレッシングとして公知である。

【0009】それぞれのガイドテーブルエントリの前記ディメンション値は、画素のアレイの水平ディメンション値および垂直ディメンション値を含む。前記メモリアクセスは、アドレスの前記ブロックからのメモリリードでありうる。前記メモリアクセスは、アドレスの前記ブロックへのメモリライトのものでありうる。実施例においては、メモリと、データプロセッサと、上述のメモリアクセスを行うデータ転送制御装置とは、単一半導体チップ内に構成されうる。該データ転送制御装置は、オンチップメモリと同様に、外部メモリにアクセスしうる。

【0010】

【実施例】図1は、本発明により画像およびグラフィック処理のために製造されたマルチプロセッサの集積回路を含む画像データ処理システムのブロック図である。このデータ処理システムは、ホスト処理システム1を含む。このホスト処理システム1は、図1のデータ処理システムのうちのホストシステムに対しデータ処理を行う。ホスト処理システム1にはプロセッサ、少なくとも一つの入力デバイス、長期記憶デバイス、リードオンリーメモリ、ランダムアクセスメモリおよびホストシステムバスに結合された少なくとも一つのホスト周辺機器が含まれる。ホスト処理システム1は、その処理機能により画像データ処理システムの機能を制御する。

【0011】マルチプロセッサ集積回路100は、図1の画像データ処理システムの画像演算のためのデータ操作および計算を含むデータ処理のほとんどを行う。マルチプロセッサ集積回路100は、画像システムバスに双方向に結合されており、この画像システムバスによりホスト処理システム1と通信するようになっている。図1の回路配列では、マルチプロセッサ集積回路100はホスト処理システム1と独立して作動する。しかしながらマルチプロセッサ集積回路100はホスト処理システム1に応答できる。

【0012】図1は2つの画像システムを示す。撮像デバイス3は画像入力デバイスとして働く文書スキャナ、電荷結合デバイススキャナまたはビデオカメラを示し、撮像デバイス3はこの画像を画像キャプチャコントローラ4へ送り、コントローラ4はこの画像をデジタル化し、画像をマスタスキャンフレームに変換するように働く。このフレームキャプチャプロセスは、マルチプロセッサ集積回路100からの信号により制御される。こうして形成された画像フレームは、ビデオランダムア

クセスメモリ5に記憶される。ビデオランダムアクセスメモリ5は、マルチプロセッサ集積回路100により画像処理のためのデータ転送を可能とする画像システムバスを介してアクセスできる。

【0013】第2画像システムはビデオディスプレイをドライブする。マルチプロセッサ集積回路100はピクセルマップを介して、ディスプレイされる画像を指定するビデオランダムアクセスメモリ6と通信する。マルチプロセッサ集積回路100は画像システムバスを介し、ビデオランダムアクセスメモリに記憶された画像データを制御する。この画像に対応するデータはビデオランダムアクセスメモリ6から再コールされ、ビデオパレット7に供給される。ビデオパレット7は、この再コールされたデータを別のカラー空間に変換し、ピクセルごとのビット数等を拡張できる。この変換はルックアップテーブルによって行うことができる。ビデオパレット7はビデオディスプレイ8を駆動するための適当なビデオ信号も発生する。これらビデオ信号がアナログ信号であれば、ビデオパレット7は適当なデジタル-アナログ変換機能も含む。このビデオパレット7から出力されるビデオレベル信号は、カラー、飽和および輝度情報を含むことができる。マルチプロセッサ集積回路100は、ビデオパレット7内に記憶されたデータを制御し、データ変換プロセスおよび画像フレームのタイミングを制御する。マルチプロセッサ集積回路100は、ビデオパレット7の制御によりビデオディスプレイ画像のうちのフレーム当たりのライン長さおよびライン数、同期信号およびリトレース信号およびブランキング信号を制御できる。重要なことに、マルチプロセッサ集積回路100は、グラフィックディスプレイ情報をビデオランダムアクセスメモリ6内のどこに記憶するかを決定し、制御する。その後ビデオランダムアクセスメモリ6からの読み出しの間、マルチプロセッサ集積回路100はビデオランダムアクセスメモリ6からの読み出しシーケンス、アクセスすべきアドレスおよびビデオディスプレイ8に所望のグラフィック画像を発生するのに必要な制御情報を決定する。

【0014】ビデオディスプレイ8は、ユーザーによって見ることができるように、指定されたビデオデータを発生する。これには2つの技術が広く使用されている。第1の技術は、各ピクセルにおける色、色合い、輝度および飽和度の点でビデオデータを特定するものであり、第2の技術では、各ピクセルに対し、赤、青および緑のカラーレベルを指定する。ビデオディスプレイ8のためのビデオパレット7は、所定の技術とコンバーチブルとなるように設計され、製造されている。図1は、画像システムバスに結合された付加メモリ9を示している。この追加メモリは、付加ビデオランダムアクセスメモリ、ダイナミックランダムアクセスメモリ、スタティックランダムアクセスメモリまたはリードオンリーメモリを

含むことができる。マルチプロセッサ集積回路100はその全体または一部をメモリ9に記憶されたプログラムにより制御できる。このメモリ9も、種々のタイプのグラフィック画像データを記憶できる。更にマルチプロセッサ集積回路100は、ビデオランダムアクセスメモリ、ダイナミックランダムアクセスメモリおよびスタティックランダムアクセスメモリのためのメモリインターフェース回路を含むことが好ましい。このように、ビデオランダムアクセスメモリ5または6を用いることなく、マルチプロセッサ集積回路100を使用してシステムを構築できる。

【0015】図1はトランシーバ16を示している。このトランシーバ16は、画像システムバスと通信チャンネルとの間で変換と双方向の通信とを行うものである。このトランシーバ16を用いたシステムの一例としてビデオ会議がある。図1に示された画像データ処理システムは、第1地点にいる人物のビデオ画像を形成するのに、撮像デバイス3と、画像キャプチャコントローラ4を用いる。マルチプロセッサ集積回路100はビデオ圧縮を行い、トランシーバ16および通信チャンネルを介して別の地点にある同様な画像データ処理システムへこの圧縮したビデオ信号を送信する。トランシーバ16は、通信チャンネルを介して遠隔地の画像データ処理システムからの同じように圧縮されたビデオ信号を受信する。マルチプロセッサ集積回路100はこの受信した信号をデコンプレスし、ビデオランダムアクセスメモリ6およびビデオパレット7を制御して、ビデオディスプレイ8上に対応するデコンプレスされたビデオ信号をディスプレイする。本発明は、画像データ処理システムがトランシーバ16を利用するような例のみに限定されるものでないことに留意されたい。更に双方向の通信は同じタイプの信号である必要はないことに留意されたい。例えば対話型ケーブルテレビ信号では、ケーブルシステムのヘッドが圧縮ビデオ信号を通信チャンネルを介して画像データ処理システムへ送り、画像データ処理システムは制御およびデータ信号をトランシーバ16および通信チャンネルを介してケーブルシステムヘッドへ送り返すことができる。

【0016】図1は、ホスト処理システム1を含むシステム内で具現化されたマルチプロセッサ集積回路100を示す。当業者であれば、マルチプロセッサ集積回路100を有効なシステムのうちの単なるプロセッサとしても使用できるような本発明の好ましい実施例を、開示内容から実現できよう。かかるシステムでは、マルチプロセッサ集積回路100はシステムの機能すべてを実行するようプログラムされる。このマルチプロセッサ集積回路100は画像処理に使用されるシステムで特に有効である。マルチプロセッサ集積回路100は複数の同一プロセッサを含むことが好ましい。これらプロセッサの各々をデジタル画像／グラフィックプロセッサと称す。こ

のような記載は、単に便宜的なものである。本発明を実施したプロセッサは、一つの集積回路または複数の集積回路上に別個に製造したプロセッサとすることができる。単一集積回路上に製造する場合、このような単一集積回路はデジタル画像／グラフィックプロセッサにより使用されるリードオンリーメモリおよびランダムアクセスメモリをオプションとして含むことができる。

【0017】図2は、マルチプロセッサ集積回路100のアーキテクチャを示す。マルチプロセッサ集積回路100は、2つのランダムアクセスメモリ10および20（その各々は複数の部分に別れている）と、クロスバー50と、マスタープロセッサ60と、デジタル画像／グラフィックプロセッサ71、72、73および74と、システムメモリへのアクセスを仲介する転送コントローラ80と、独立した第1および第2画像メモリへのアクセスを制御できるフレームコントローラ90とを含む。マイクロプロセッサ集積回路100は、高度のオペレーションパラレリズム（並行性）を提供する。これは画像処理およびグラフィックオペレーション、例えばマルチメディアの計算において有効である。これらプロセッサが有効となる画像およびグラフィック処理以外の計算アプリケーションもあるので、画像／グラフィックプロセッサとしてプロセッサ71、72、73および74を参考としたことは、単なる便宜的なものである。

【0018】マイクロプロセッサ集積回路100は2つのランダムアクセスメモリを含む。ランダムアクセスメモリ10は、主にマスタープロセッサ60の専用である。このメモリは2つのインストラクションキャッシュメモリ11および12と、2つのデータキャッシュメモリ13および14と、パラメータメモリ15を含む。これらメモリ部分は物理的には同一とすることができるが、異なるように接続し、使用することもできる。ランダムアクセスメモリ20はマスタープロセッサ60およびデジタル画像／グラフィックプロセッサ71、72、73および74の各々によってアクセスできる。各デジタル画像／グラフィックプロセッサ71、72、73および74は、5つの対応するメモリ部分を有する。これらはインストラクションキャッシュメモリ、3つのデータメモリおよび1つのパラメータメモリを含む。従ってデジタル画像／グラフィックプロセッサ71は対応するインストラクションキャッシュメモリ21、データメモリ22、23、24およびパラメータメモリ25を有し、デジタル画像／グラフィックプロセッサ72は対応するインストラクションキャッシュメモリ26と、データメモリ27、28、29およびパラメータメモリ30を有し、デジタル画像／グラフィックプロセッサ73は対応するインストラクションキャッシュメモリ31と、データメモリ32、33、34およびパラメータメモリ35を有し、デジタル画像／グラフィックプロセッサ74は対応するインストラクションキャッシュメモリ36

と、データメモリ37、38、39およびパラメータメモリ40とを有する。ランダムアクセスメモリ10の部分と同じように、これらメモリ部分は物理的に同一にできるが、異なるように接続し、使用することもできる。メモリ10および20のうちのこれらメモリ部分の各々は、例えば20Kバイトを含み、マルチプロセッサ集積回路100内の総メモリは50Kバイトとなる。

【0019】マルチプロセッサ集積回路100は、複数の独立したパラレルデータ転送を用いることにより、プロセッサとメモリとの間で高レートでのデータ転送を行うように製造されている。クロスバー50はこれらのデータ転送を可能にするものであり、各デジタル画像/グラフィックプロセッサ71、72、73および74は、サイクルごとに同時に作動できる3つのメモリポートを有する。インストラクションポート(I)は対応するインストラクションキャッシュから64ビットのデータワードをフェッチでき、ローカルデータポート(L)はデジタル画像/グラフィックプロセッサに対応するデータメモリまたはパラメータメモリから32ビットのデータワードを読み出したり、これに書き込みを行うことができる。グローバルデータポート(G)は、データメモリまたはパラメータメモリまたはランダムアクセスメモリ20のいずれかから32ビットのデータワードを読み出したり、書き込んだりできる。マスタープロセッサ60は、2つのメモリポートを含む。インストラクションポート(I)はインストラクションキャッシュ11および12のいずれかから32ビットのインストラクションワードをフェッチできる。データポート(C)はデータキャッシュ13または14、ランダムアクセスメモリ10のうちのパラメータメモリ15、またはデータメモリ、パラメータメモリ、ランダムアクセスメモリ20の任意のものから32ビットのデータワードを読み出したり、これらに書き込みできる。転送コントローラ80はデータポート(C)を介してランダムアクセスメモリ10または20の部分のいずれかにアクセスできる。従って一つのメモリサイクルで16のパラレルメモリアクセスをリクエストできる。このような多数のパラレルアクセスをサポートするように、ランダムアクセスメモリ10および20は、25のメモリに分割されている。

【0020】クロスバー50はマスタープロセッサ60、デジタル画像/グラフィックプロセッサ71、72、73および74、および転送コントローラ80とメモリ10および20との接続を制御する。クロスバー50は、列と行に配置された複数のクロスポイント51を含む。クロスポイント51の各列は、単一メモリ部分および対応するアドレスのレンジに対応する。プロセッサはこのプロセッサが出力するアドレスの最高位ビットによるメモリ部分のうちの一つへのアクセスをリクエストする。プロセッサにより出力されるこのアドレスは、行に沿って進む。このアドレスを有するメモリ部分に対応

するクロスポイント51は、メモリ部分へのアクセスを許可または否定することによって応答する。それ以外のいずれのプロセッサも、そのときのメモリサイクルの間にそのメモリ部分へのアクセスをリクエストしない場合、クロスポイント51が行と列を結合することにより、アクセスを許可する。これによりそのメモリ部分にアドレスが供給される。このメモリ部分はそのアドレスにおけるデータアクセスを可能にすることによって応答する。このデータアクセスはデータ読み出しオペレーションまたはデータ書き込みオペレーションのいずれかでよい。

【0021】2つ以上のプロセッサが同じメモリ部分へのアクセスを同時にリクエストする場合、クロスバー50はリクエスト中のプロセッサのうちの一つのアクセスしか許可しない。クロスバー50のうちの各列のうちのクロスポイント51は、優先階層に基づいて通信し、アクセスを許可する。同じランクを有するアクセスの2つのリクエストが同時に行われた場合、クロスバー50は最後に許可されたプロセッサが最も低い優先度を有するラウンドロビン法によりアクセスを許可する。リクエストにサービスする必要がある限り、各々の許可されたアクセスが続く。プロセッサはメモリサイクルごとにアドレスを変更できるので、クロスバー50はサイクルごとにプロセッサとメモリ部分との間の相互接続を変更できる。マスタープロセッサ60は、マルチプロセッサ集積回路100のための主要制御機能を実行することが好ましい。マスタープロセッサ60はハードウェアの浮動小数点計算ユニットを含む32ビットの縮小インストラクションセットコンピュータ(RISC)プロセッサであることが好ましい。RISCアーキテクチャによれば、メモリへのすべてのアクセスは、ロードおよび記憶インストラクションで実行され、ほとんどの整数および論理演算はレジスタ上で1回のサイクルで実行される。しかしながら整数および論理ユニットによって使用されるのと同じレジスタファイルを用いる場合、一般に演算を実行するには2サイクルがかかる。レジスタのスコアボードは正しいレジスタアクセスシーケンスを維持するように保証するものであり、画像処理における制御機能に対してはRISCアーキテクチャが適当である。浮動小数点計算ユニットは画像回転機能の高速計算を可能にするもので、このような機能は画像処理に重要である。

【0022】マスタープロセッサ60は、インストラクションキャッシュメモリ11またはインストラクションキャッシュメモリ12からのインストラクションワードをフェッチする。同じように、マスタープロセッサ60はデータキャッシュ13またはデータキャッシュ14のいずれかからデータをフェッチする。各メモリ部分は2Kバイトのメモリを含むので、14Kバイトのインストラクションキャッシュと4Kバイトのデータキャッシュがあることになる。キャッシュ制御はマスタープロセッサ60の不可欠な機能であり、上記のようにマスタープ

ロセッサ60はクロスバー50を介して他のメモリ部分にもアクセスできる。4つのデジタル画像／グラフィックプロセッサ71、72、73および74の各々は、高度にパラレルなデジタル信号プロセッサ(DSP)アーキテクチャを有する。デジタル画像／グラフィックプロセッサ71、72、73および74は3つの別個のユニット、すなわちデータユニットと、アドレスユニットと、プログラムフロー制御ユニットを用いた高度なオペレーションのパラレリズムが得られる。これら3つのユニットは、インストラクションパイプライン内の異なる命令で同時に作動する、更にこれらユニットの各々は、内部パラレリズムを含む。

【0023】デジタル画像／グラフィックプロセッサ71、72、73および74は、マルチインストラクションマルチデータモード(MIMD)で独立したインストラクションストリームを実行できる。このMIMDモードでは、各デジタル画像／グラフィックプロセッサは、対応するインストラクションキャッシュからの個々のプログラム(これは独立的でも協働的でもよい)を実行する。後者のケースでは、クロスバー50は共用メモリと共に、プロセッサ内通信を可能にする。デジタル画像／グラフィックプロセッサ71、72、73および74は、同期MIMDモードでも作動できる。同期MIMDモードでは、各デジタル画像／グラフィックプロセッサのプログラムフロー制御ユニット130は、すべての同期したプロセッサが進む準備がなされるまで、次のインストラクションのフェッチを禁止する。このような同期MIMDモードは密に結合されたオペレーションにおけるロックステップでデジタル画像／グラフィックプロセッサの別々のプログラムを実行できるようにするものである。

【0024】デジタル画像／グラフィックプロセッサ71、72、73、74は、単一インストラクションマルチデータモード(SIMD)で異なるデータに対する同じインストラクションを実行できる。このモードにおいて、4つのデジタル画像／グラフィックプロセッサのための単一インストラクションストリームは、インストラクションキャッシュメモリ21から生じる。データユニット画像／グラフィックプロセッサ71は、フェッチオペレーションおよびブランチオペレーションを制御し、クロスバー50は他のデジタル画像／グラフィックプロセッサ72、73および74へ同じインストラクションを供給する。デジタル画像／グラフィックプロセッサ71は、デジタル画像／グラフィックプロセッサ71、72、73および74のすべてに対するインストラクションフェッチを制御するので、これらデジタル画像／グラフィックプロセッサは本来SIMDモードでは同期化されるものである。

【0025】転送コントローラ80は、マルチプロセッサ集積回路100のための組み合わせされたダイレクトメ

モリアクセス(DMA)マシンと、メモリインターフェースであり、この転送コントローラ80は、インテリジェントにキューイングし、優先度をセットし、5つのプログラマブルプロセッサのデータリクエストおよびキャッシュミスにサービスする。マスタープロセッサ60およびデジタル画像／グラフィックプロセッサ71、72、73および74のいずれも、転送コントローラ80を介してマルチプロセッサ集積回路100の外部のメモリおよびシステムにアクセスする。データキャッシュまたはインストラクションキャッシュミスは、転送コントローラ80によって自動的に取り扱われる。キャッシュサービス(S)ポートは、かかるキャッシュミスを転送コントローラ80へ伝送し、キャッシュサービスポート(S)はメモリからではなくプロセッサから情報を読み出すマスタープロセッサ60およびデジタル画像／グラフィックプロセッサ71、72、73および74は、リンクされたリストパケット転送として転送コントローラ80からのデータ転送をリクエストできる。これらリンクされたリストパケット転送は、ソースメモリアドレスと宛て先メモリアドレスとの間で多次元ブロックの情報の転送を可能にするものであり、これらアドレスはマルチプロセッサ集積回路100内にあってもよいし、またはマルチプロセッサ集積回路100の外部にあってもよい。転送コントローラ80は、内部のデータを保持するのに周期的なリフレッシュを必要とするダイナミックランダムアクセスメモリ(DRAM)用のリフレッシュコントローラも含むことが好ましい。

【0026】フレームコントローラ90はマルチプロセッサ集積回路100と、外部画像キャプチャおよびディスプレイシステムとの間のインターフェースである。このフレームコントローラ90は、キャプチャおよびディスプレイデバイスに対する制御を行い、これらデバイスとメモリとの間のデータの移動を自動的に管理する。このため、フレームコントローラ90は、2つの独立した画像システムに対して同時に制御を行う。これら画像システムは一般に画像キャプチャ(取り込み)用第1画像システムと画像ディスプレイ用の第2画像システムとからなるが、フレームコントローラ90の使用はユーザーによって制御される。これら画像システムは、通常、フレームグラバーまたはフレームバッファ記憶装置のいずれかに対して使用される独立したフレームメモリを含む。フレームコントローラ90は、リフレッシュおよびシフトレジスタ制御により、ビデオダイナミックランダムアクセスメモリ(VRAM)を制御するように作動することが好ましい。

【0027】マルチプロセッサ集積回路100は、大規模画像処理のために設計されている。マスタープロセッサ60は埋め込み制御を行い、デジタル画像／グラフィックプロセッサ71、72、73、74の活動を調和させ、これらプロセッサの発生した結果を解釈する。デジ

タル画像／グラフィックプロセッサ71、72、73、74は、ピクセル解析および操作に良好に適している。ピクセルのデータが多く、情報が少ないとみなされる場合、代表的なアプリケーションではデジタル画像／グラフィックプロセッサ71、72、73、74がピクセルを良好に検査し、未加工データを情報とする。次にこの情報は、デジタル画像／グラフィックプロセッサ71、72、73、74またはマスタープロセッサ60のいずれかによって分析できる。クロスバー50はプロセッサ内通信を仲介する。更にこのクロスバー50はマスタープロセッサ集積回路100を共用メモリシステムとして実現できるようにする。このアーキテクチャでは、メッセージの通過は通信の主な形態となる必要はない。しかしながら共用メモリをメッセージが通過するようにできる。各デジタル画像／グラフィックプロセッサ、クロスバー50の対応する部分およびメモリ20の対応する部分は、この実施例内で同じ幅を有する。これにより、同じピン出力を維持しながら、モジュラー式のデジタル画像／グラフィックプロセッサおよび対応するメモリの追加または取り外しに適応できるようにすることにより、アーキテクチャをフレキシブルにできる。

【0028】ある実施例では、マルチプロセッサ集積回路100のすべての部品は、0.6 μ mの特徴的寸法を用いた相補的酸化金属半導体(CMOS)に形成された単一集積回路に配置されている。マルチプロセッサ集積回路100は、256個のピンを有するピングリッドアレイパッケージ内に適当に製造される。入出力端は、例えばTTL論理電圧にコンパチブルであり、マルチプロセッサ集積回路100は、約300万個のトランジスタを含み、50MHzのクロックレートを使用している。図3は、インタラプトイネーブルレジスタINTEN110およびインタラプトフラグレジスタINTFLG115のためのフィールド定義を示す。rと表示されたビットは、将来の使用に対して保留されており、一のついたビットは、好ましい実施例では使用されないが、別の実施例では使用できる。インタラプト(割り込み)は、左から右に優先度が決められている。各インタラプトソースは、インタラプトイネーブルレジスタINTEN110の対応するイネーブル(E)ビット内に1をセットすることによって、個々にイネーブルできる。インタラプトフラグレジスタINTFLG115のインタラプトソースビットは、右から左へ、すなわち常時イネーブルにされているエミュレーションインタラプトETRAP、XYパッチインタラプト、タスクインタラプト、パケット転送ビジーインタラプトPTB、パケット転送エラーインタラプトPTERROR、パケット転送成功インタラプトPTEND、マスタープロセッサ60メッセージインタラプトMPMSG、デジタル画像／グラフィックプロセッサ71メッセージインタラプトDIGPOMS、デジタル画像／グラフィックプロセッサ72メッ

ージインタラプトDIGP1MSG、デジタル画像／グラフィックプロセッサ73メッセージインタラプトDIGP2MSG、デジタル画像／グラフィックプロセッサ74メッセージインタラプトDIGP3MSGの方向に、優先度が低くなるようになっている。ビット31～28は、8個のデジタル画像／グラフィックプロセッサを含むマルチプロセッサ集積回路100の実現の際に4つの追加デジタル画像／グラフィックプロセッサからのメッセージインタラプトのために保留されている。

【0029】インタラプトイネーブルレジスタINTEN110のWビット(ビット0)は、インタラプトフラグレジスタINTFLG115への書き込みを制御する。通常、このビットはエミュレーションインタラプトをイネーブルするかどうかを制御する。好ましい実施例では、エミュレーションインタラプトはディスエーブルできないので、インタラプトイネーブルレジスタINTEN110内で、このインタラプトに対するイネーブルビットの必要はない。インタラプトイネーブルレジスタINTEN110のビット0は、インタラプトフラグレジスタINTFLG115の作動を変えるものである。インタラプトイネーブルレジスタINTEN110のWビットが1であると、インタラプトフラグレジスタINTFLG115へのソフトウェアの書き込みはビットを1にセットできるだけである。これらの条件では、インタラプトフラグレジスタINTFLG115のビットへの0の書き込みは全く効果はない。このようなWビットが0である場合、インタラプトフラグレジスタINTFLG115のいずれかのビットへの1の書き込みは、そのビットを0にクリアする。インタラプトフラグレジスタINTFLG115の任意のビットへの0への書き込みは全く効果がない。これにより、他のステートを乱すことなくインタラプトフラグレジスタINTFLG115内の個々のインタラプトフラグをクリアできる。各インタラプトサービスルーチンは、リターン前に対応するインタラプトフラグを適当にクリアするが、その理由は好ましい実施例ではハードウェアによってこれらフラグをクリアしていないからである。この唯一の例外として、エミュレーションインタラプトETRAPハードウェアによってクリアされるが、この理由は、このようなインタラプトが常時イネーブルされているからである。特定のインタラプトソースがソフトウェアの書き込みによってこれをクリアするのと同じように、インタラプトフラグレジスタINTFLG115内のビットを同時にセットしようとしている場合、ロジックによりこのビットはセットされる。

【0030】ETRAPインタラプトフラグ(インタラプトフラグレジスタINTFLG115のビット0)は、解析論理またはETRAPインストラクションのいずれかからセットされる。このインタラプトはディスエーブルされないで、通常は即座にサービスされるが、

インタラプトサービスはパイプラインストール条件、例えばクロスバー50を介するメモリの競合が解決されるまで待機する。ENTRAPインタラプトフラグは、インタラプトサービスがされる際にハードウェアによってクリアされるインタラプトフラグレジスタINTFLG115内の単なるインタラプトビットである。XY PATCHインタラプトフラグ（インタラプトフラグレジスタINTFLG115のビット11）は、XYアドレス指定をするのにグローバルアドレスユニット610と、ローカルアドレスユニット620との組み合わせを用いる際に、所定の条件でセットされる。XYパッチされたアドレス指定は所定の条件でインタラプトを発生できる。XYパッチされたアドレス指定のためのインストラクションワードの呼び出しは、かかるインタラプトを発生できるかどうか、更に指定されたパッチの内外でアドレス上で許可されたインタラプトを行うかどうかを示している。

【0031】マスタプロセッサ60からのコマンドを受信した際に、TASKインタラプトフラグ（インタラプトフラグレジスタINTFLG115内のビット14）がセットされる。このインタラプトによりデジタル画像／グラフィックプロセッサ71がそのTASKインタラプトベクトルをロードする。このインタラプトは、例えばマスタプロセッサ60の制御により、選択されたデジタル画像／グラフィックプロセッサ71、72、73、74をスイッチングさせることができる。キューアクティブビットが1のとき、ソフトウェアが通信用レジスタCOMM120のパケット転送ビットに1を書き込む場合、パケット転送ビジーインタラプトフラグPTB（インタラプトフラグレジスタINTFLG115のビット17）がセットされる。これにより先のパケット転送が終了したかどうかをチェックすることなくパケット転送を送ることができる。先のパケット転送がまだ待機中であれば、このインタラプトフラグはセット状態となる。これについては通信用レジスタCOMM120の説明と関連して後に詳細に述べる。

【0032】デジタル画像／グラフィックプロセッサにより送られるパケット転送を実行する間に、転送用コントローラ80がエラー条件に遭遇すれば、パケット転送エラーインタラプトフラグPTERROR（インタラプトフラグレジスタINTFLG115のビット18）がセットされる。転送コントローラ80がデジタル画像／グラフィックプロセッサのリンクされたリストの終了部に遭遇するか、または終了時にリクエスト中のデジタル画像／グラフィックプロセッサをインタラプトするよう、転送コントローラ80に命令するパケット転送を完了した際に、パケット転送エンドインタラプトフラグPTEND（インタラプトフラグレジスタINTFLG1

15のビット19）が転送コントローラ80によってセットされる。

【0033】マスタプロセッサ60がそのデジタル画像／グラフィックプロセッサにメッセージインタラプトを送ると、マスタプロセッサメッセージインタラプトフラグMPMSG（インタラプトフラグレジスタINTFLG115のビット20）がセット状態となる。インタラプトフラグレジスタINTFLG115のビット27～24は、デジタル画像／グラフィックプロセッサ71、72、73、74からのメッセージインタラプトをログする。デジタル画像／グラフィックプロセッサ71、72、73または74は、インタラプトフラグレジスタINTFLG115の対応するビットを介してそれ自体にメッセージを送ったり、それ自体をインタラプトできる。デジタル画像／グラフィックプロセッサ71が、デジタル画像／グラフィックプロセッサにメッセージインタラプトを送ると、デジタル画像／グラフィックプロセッサ0のメッセージインタラプトフラグDIGPOMSG（インタラプトフラグレジスタINTFLG115のビット24）がセットされる。同様に、デジタル画像／グラフィックプロセッサ72が、メッセージインタラプトを送ると、デジタル画像／グラフィックプロセッサ1のメッセージインタラプトフラグDIGP1MSG（インタラプトフラグレジスタINTFLG115のビット25）がセットされる。デジタル画像／グラフィックプロセッサ73が、メッセージインタラプトを送ると、デジタル画像／グラフィックプロセッサ2のメッセージインタラプトフラグDIGP2MSG（インタラプトフラグレジスタINTFLG115のビット26）がセットされる。デジタル画像／グラフィックプロセッサ74が、メッセージインタラプトを送ると、デジタル画像／グラフィックプロセッサ3のメッセージインタラプトフラグDIGP3MSG（インタラプトフラグレジスタINTFLG115のビット27）がセットされる。先に述べたように、インタラプトフラグレジスタINTFLG115のビット31～28は、8個のデジタル画像／グラフィックプロセッサを含むマルチプロセッサ集積回路100を実現する際の4つの付加的デジタル画像／グラフィックプロセッサからのメッセージインタラプトのために保留されている。

【0034】イネーブルされたインタラプトが起きると、小ステートマシンにすることができるインタラプト疑似インストラクションユニットがインストラクションレジスタアドレスステージでパイプラインに次の疑似インストラクションの組を注入する。

【0035】

【数1】

・ (A14 - 16) = SR

・ (A14 + 12) = PC

BR = vectadd: S, GおよびLをロードするための

vectaddの2つの最小位ビット=11

・ (A14 + 8) = IPA

・ (A14 + 4) = IPE

【0036】これら疑似インストラクションをそれぞれPS1、PS2、PS3、PS4およびPS5と称す。このシーケンスによってはサブルーチンIPRSからのインストラクションポインタリターンはセーブされない。インタラプトサービスルーチンがなんらかのブランチを実行する場合、インタラプトサービスルーチンによってまず最初にサブルーチンIPRSからのインストラクションポインタリターンをブッシュし、リターン前にレストアしなければならない。ベクトルフェッチは保護されているサブルーチンIPRSからのインストラクションポインタリターンを全プログラムカウンタPC701にロードすることである。これによりプログラムカウンタPC701のS、GおよびLビットがロードされるので、すべてのインタラプトベクトルの3つの最小位ビットが0にされる。この説明の例外は、リセット後にフェッチされるタスクベクトルは、ルーピングをディスエーブルするようにLビット（プログラムカウンタPC701のビット0）をセットさせなければならないことである。

【0037】インタラプトフラグレジスタINTFLC115内に示されたインタラプトのためのインタラプトサービスルーチンの開始ポイントのそれぞれのアドレスのデジタル画像／グラフィックプロセッサインタラプトベクトルと称す。これらアドレスはソフトウェアによ

て発生され、表1に示されるそれぞれのインタラプトされたデジタル画像／グラフィックプロセッサ71、72、73、74に対応するパラメータメモリ25、30、35、40にデータとしてロードされる。インタラプト疑似インストラクションPS3は、対応するパラメータメモリ25、30、35または40における表示されたアドレスに記憶された32ビットのアドレスを取り出し、これをプログラムカウンタPC701に記憶する。インタラプト疑似インストラクションユニット770はインタラプトイネーブルレジスタによってイネーブルされた最高優先度のインタラプトに基づき、対応するパラメータメモリのためにアドレスを計算する。インタラプト疑似インストラクションユニット770は各デジタル画像／グラフィックプロセッサのためのユニークなアドレスを発生するよう、通信レジスタCOMM120からのデジタル画像／グラフィックプロセッサ番号を含むように作動する。インタラプト疑似インストラクションPS4およびPS5は、インタラプトサービスルーチンへのブランチの後のディレイスロット内にあることに留意されたい。

【0038】

【表1】

INTFLG ビット	インタラプト名	アドレス
31	DIGP7メッセージ用に保留	0100#1FC
30	DIGP6メッセージ用に保留	0100#1F8
29	DIGP5メッセージ用に保留	0100#1F4
28	DIGP4メッセージ用に保留	0100#1F0
27	DIGP3メッセージ	0100#1EC
26	DIGP2メッセージ	0100#1E8
25	DIGP1メッセージ	0100#1E4
24	DIGP0メッセージ	0100#1E0
23	スベア	0100#1DC
22	スベア	0100#1D8
21	スベア	0100#1D4
20	マスタプロセッサメッセージ	0100#1D0
19	パケット転送成功	0100#1CC
18	パケット転送エラー	0100#1C8
17	パケット転送ビジー	0100#1C4
16	スベア	0100#1C0
15	スベア	0100#1BC
14	TASKインタラプト	0100#1B8
13	スベア	0100#1B4
12	スベア	0100#1B0
11	XYバッチング	0100#1AC
10	保留	0100#1A8
9	保留	0100#1A4
8	保留	0100#1A0
7	保留	0100#19C
6	保留	0100#198
5	保留	0100#194
4	保留	0100#190
3	保留	0100#18C
2	スベア	0100#188
1	スベア	0100#184
0	エミュレーション	0100#180

表1

【0039】各アドレスにて#は通信レジスタCOMM120から得られたデジタル画像／グラフィックプロセッサ番号と置換される。

【0040】インタラプトサービスルーチンの最後の4つのインストラクションは、次の(32ビットデータのシフトされていないインデックス)オペレーションを含んでいなければならない。

【0041】

【数2】

SR = * (A14 ++= 4)

BR = * (A14 ++= 7)

BR = * (A14 ++= 5)

BR = * (A14 ++= 5)

【0042】これらインストラクションをそれぞれRETI1、RETI2、RETI3およびRETI4と称す。他のオペレーションは希望すればこれらオペレーションと平行にコード化できるが、これらのオペレーションのいずれもステータスレジスタ211を変えてはならない。

【0043】新しいタスクをデジタル画像／グラフィックプロセッサ上で実行すべきであり、新しいタスクを終了した後に最初のステートに復帰すべき場合インタラプトステートをセーブできる。インタラプトイネーブルレ

ジスタINTEN110上のWビットにより制御される書き込みモードにより、セービングまたはレストアオペレーション中にインタラプトを失うことなくこれを行うことができる。これは次のインストラクションシーケンスで実行できる。まずDINTインストラクションによりインタラプトをディスエーブルし、次にインタラプトイネーブルレジスタINTEN110およびインタラプトフラグレジスタINTFLG115の双方をセーブし、インタラプトイネーブルレジスタINTEN110のWビット(ビット0)を0にセットし、16進数FFFFFFFをインタラプトフラグレジスタINTFLG115に書き込む。イネープリングインタラプトを含むことができる新しいタスクを実行する。新しいタスクの完了後、元のタスクをリカバーし、最初にDINTインストラクションによりインタラプトをディスエーブルし、インタラプトイネーブルレジスタINTEN110のWビットを1にセットする。メモリからインタラプトフラグレジスタINTFLG115のステータスをレストアし、次にメモリからインタラプトイネーブルレジスタINTEN110のステータスをレストアする。最後にEINTインストラクションによりインタラプトをイネーブルする。

【0044】各デジタル画像／グラフィックプロセッサ71、72、73、74は、他のデジタル画像／グラフィックプロセッサおよびマスタプロセッサ60にコマン

ドワードを送信できる。レジスタA15の宛て先、グローバルアドレスユニットのうちの0値のアドレスレジスタによるレジスタ間の移動により、宛て先プロセッサへのコマンドワードの転送が開始される。このようなレジスタ間の転送は、後述するように、一つのインストラクションでデータユニット110のオペレーションおよびローカルデータポート144を介したアクセスと組み合わせることができる。このようなコマンドワードは特別コマンドワード信号に伴われてグローバルデータポート148を介してクロスバー50へ送信される。これによりマスタプロセッサ60およびデジタル画像／グラフィックプロセッサ71、72、73、74はマルチプロセッサ集積回路100の他のプロセッサとの通信が認められる。

【0045】図4は、これらコマンドワードのフィールド定義を略図で示している。好ましい実施例では、コマンドワードはグローバルデータポート148を介して送信されるデータと同一の32ビット長さを有する。各コマンドワードの最小位のビットは、コマンドワードがアドレス指定される一つ以上のプロセッサおよび他の回路を定義している。各受け入れ側回路は、これらビットがコマンドワードをその回路に向けることを表示している場合に限り、受信したコマンドワードに応答する。各コマンドワードのビット3～0はそれぞれデジタル画像／グラフィックプロセッサ74、73、72、71をそれぞれ指定している。好ましい実施例ではビット7～4は使用されず、8個のデジタル画像／グラフィックプロセッサを有するマルチプロセッサ集積回路100内で使用するために保留されている。ビット8はマルチプロセッサ60にコマンドワードをアドレス指定することを表示している。ビット9はコマンドワードを転送コントローラ80に向けることを表示しており、ビット10はコマンドワードをフレームコントローラ90へ向けることを表示している。ここですべての回路はすべてのコマンドワードを他のすべての回路に送ることを許可されているわけではないことに留意されたい。例えばシステムレベルのコマンドワードはデジタル画像／グラフィックプロセッサから他のデジタル画像／グラフィックプロセッサまたはマスタプロセッサ60へ送ることはできない。マスタプロセッサ60しか、転送コントローラ80またはフレームコントローラ90へコマンドワードを送ることができない。どの回路がどのコマンドワードを他のどの回路に送ることができるかの制限について、各コマンドワードフィールドの説明と関連させて後に説明する。

【0046】コマンドワードのRビット（ビット31）は、リセットビットである。マスタプロセッサ60はこのワードをデジタル画像／グラフィックプロセッサに対して発生したり、またはあるデジタル画像／グラフィックプロセッサがこのコマンドワードを自己に発生したりできる。意図している実施例は、いずれのデジタル画像

／グラフィックプロセッサの他のデジタル画像／グラフィックプロセッサをリセットできない。下記のリセットシーケンスの説明全体で、アドレス内の各数字の#はコマンドレジスタCOMM120のビット1～0に記憶されたデジタル画像／グラフィックプロセッサの番号と置換すべきであることに留意されたい。指定されたデジタル画像／グラフィックプロセッサがリセットコマンドワードを受けると、まずホールドラッチをセットし、リセットリクエスト信号を転送コントローラ80へ送る。転送コントローラ80はリセットアクノリッジ信号をデジタル画像／グラフィックプロセッサへ送る。リセット中のデジタル画像／グラフィックプロセッサは、転送プロセッサ80からリセットアクノリッジ信号が受信されるまで、別の動作は実行しない。リセットアクノリッジの受信時にデジタル画像／グラフィックプロセッサは次のシーケンスのオペレーションを開始する。すなわち既にセットされていなければホールドラッチをセットし、通信レジスタCOMM120のF、P、QおよびSビットを0にクリア（これらビットの使用については後述する）し、アドレスユニット120によりペンディング中のメモリアccessをクリアし、インストラクションキャッシュサバスリクエストをリセットし、インストラクションレジスタ実行ステージIRE752に次のインストラクションをロードする。

【0047】

【数3】

BR = [u.ncvz] A14 < 1

⌈ A14 = Hex "0100"7FC

【0048】このインストラクションはプログラムカウンタPC701に対して1ビット左にシフトされたスタックポインタA14の内容に対して変化しないようにされている否定、桁上げ、オーバーフローおよび0ステータスビット、更にRビットセットを無条件にロードし、スタックポインタA14のロードと並行にスタックポインタA14をリセットし、インストラクションレジスタアドレスステージIRAに下記のインストラクションをロードする。

【0049】

【数4】

*(PBA + Hex "FC") = PC

【0050】このインストラクションはアドレスPBAと16進数FCの合計によって表示されるアドレスにプログラムカウンタPC701の内容を記憶させ、インタラプト疑似インストラクションユニット770をセットして、次にインタラプト疑似インストラクションPS3をロードし、タスクインタラプトを表示するインタラプトフラグレジスタINTFLG115のビット14をセットし、インタラプトフラグレジスタINTFLG11

5のビット0をクリアしてエミュレータトラップインタラプトETRAPをクリアし、ループ制御レジスタLC TLのビット11、7および3をクリアして、よってすべてのループをディスエーブルする。

【0051】マスタプロセッサ60がアンホールドコマンドワードを送信すると、デジタル画像／グラフィックプロセッサによる実行が開始する。この実行によってデジタル画像／グラフィックプロセッサの作動が開始すると16進数010#7FCをアドレス指定するよう、プログラムカウンタPC701に記憶されていたアドレスをセーブする。これにより12ビットだけ左にシフトされていたスタックポインタA14の前の内容およびプログラムカウンタPC701の制御ビット（ビット2～0）の現在の値がセーブされ、アドレス16進数010*

*0#7F0をスタックポインタA14にロードし、プログラムカウンタPC701に制御ビット2～0が000となっているタスクインタラプトベクトルをロードし、アドレス16進数0100#7F8に制御ビット2～0を含むインストラクションレジスタアドレスステージIPAの内容を記憶し、アドレス16進数0100#7F4に制御ビット2～0を含むインストラクションレジスタ実行ステージIREの内容を記憶し、タスクインタラプトによって示されたアドレスにおいてプログラムの実行を開始する。表2にリセット後のスタックステートが示されている。

【0052】

【表2】

アドレス	内容
16進数0100#7FC	リセット前の左に1つシフトされた場所からのスタックポインタレジスタA14
16進数0100#7F8	リセット前からのインストラクションレジスタアドレスステージIRA
16進数0100#7F4	リセット前からのインストラクションレジスタ実行ステージIRE

表2

【0053】インストラクションレジスタアドレスステージIRAおよびインストラクションレジスタ実行ステージIREの先のステートは、制御ビット2～0を含む。スタックポインタA14はアドレス16進数0100#7F0を含むことに留意のこと。

【0054】コマンドワードのうちのHビット（ビット30）はホールドビットである。マスタプロセッサ60はこのコマンドワードを任意のデジタル画像／グラフィックプロセッサに発生することもできるし、あるデジタル画像／グラフィックプロセッサはこのコマンドを自らに発生することもできる。意図している実施例では、いずれのデジタル画像／グラフィックプロセッサも他のデジタル画像／グラフィックプロセッサを停止することはできない。指定されたデジタル画像／グラフィックプロセッサがこのコマンドワードを受信すると、デジタル画像／グラフィックプロセッサはホールドラッチをセットし、パイプラインを停止させる。この後のデジタル画像／グラフィックプロセッサは無限にクロスバメモリが競合しているかのように働く。何もリセットされず、何もインタラプトは起きないか、またはそのように認識される。あるデジタル画像／グラフィックプロセッサはコマンドワードを送ることにより自身を停止させると、ホールドコマンドワードを送るインストラクションの後の2つのインストラクションは、そのインストラクションパイプライン内にあることに留意されたい。ホールドコマンドワードを発生したインストラクションの後の第1インストラクションのアドレスパイプラインステージが

インストラクションパイプラインの性質により、すでにそのアドレスパイプラインステージを実行することになることに留意されたい。このようなホールドステートはマスタプロセッサ60からのアンホールドコマンドワードを受信することによってしか反転できない。

【0055】ホールド条件は変わらないので、デジタル画像／グラフィックプロセッサ内のパワー消費量を減少させる。デジタル画像／グラフィックプロセッサがこのモードになっている間、クロックを停止することにより更にパワーを節約できる。

【0056】コマンドワードのUビット（ビット29）は、アンホールドビットである。このコマンドワードはマスタプロセッサ60から一つ以上のデジタル画像／グラフィックプロセッサ71、72、73、74によってしか発生できない。アンホールドコマンドワードは宛て先デジタル画像／グラフィックプロセッサのホールドラッチをクリアし、デジタル画像／グラフィックプロセッサは次に、何も起こらなかったがごとくホールドの後にコードの実行を再開する。これはハードウェアまたはコードリセットの後でデジタル画像／グラフィックプロセッサをスタートさせる好ましい方法である。アンホールドコマンドワードの実行時に宛て先デジタル画像／グラフィックプロセッサは、そのタスクインタラプトベクトルによって示されたアドレスでもコードの実行を始める。Uビットの優先度は単一コマンドワードのHビットよりも高い。したがってHビットとUビットとの双方を備えた単一コマンドワードを受信する結果、アンホール

トコマンドが実行される。マスタプロセッサ60からのアンホールドコマンドワードとデジタル画像／グラフィックプロセッサ自体により送信されるホールドコマンドワードを同時に受信することにより、マスタプロセッサ60のアンホールドコマンドワードに優先権が与えられる。これによりRビットはUビットより優先度が高くなる。従って、RビットとUビットセットの双方を有するマスタプロセッサ60からの単一コマンドワードが受信される結果、デジタル画像／グラフィックプロセッサはホールド状態にリセットされる。

【0057】コマンドワードのIビット（ビット28）は、インストラクションキャッシュフラッシュビットである。マスタプロセッサ60がこのコマンドワードをいずれかのデジタル画像／グラフィックプロセッサに送ってもよいし、一つのデジタル画像／グラフィックプロセッサがかかるコマンドワードを自身に発生してもよい。意図している実施例では、いずれのデジタル画像／グラフィックプロセッサも他のデジタル画像／グラフィックプロセッサによるインストラクションキャッシュフラッシュを命令できない。このコマンドワードを受信する指定されたデジタル画像／グラフィックプロセッサはそのインストラクションキャッシュをフラッシュする。インストラクションキャッシュフラッシュは、キャッシュタグの値フィールドをキャッシュタグレジスタ自身の番号にセットさせ、存在するビットのすべてをクリアし、L、R、Uビットをタグレジスタ自身の番号にセットする。

【0058】コマンドワードのDビット（ビット27）は、データキャッシュフィールドを表示する。デジタル画像／グラフィックプロセッサ71、72、73、74は、データキャッシュを使用しないので、このコマンドワードはデジタル画像／グラフィックプロセッサに適用されず、これらに無視される。マスタプロセッサ60はこのコマンドワードを自らに送り、そのデータキャッシュメモリ13および14をフラッシュさせることができる。

【0059】コマンドワードのKビット（ビット14）は、タスクインタラプトを表示している。マスタプロセッサ60はこのコマンドワードをいずれのデジタル画像／グラフィックプロセッサ71、72、73または74にも送ることができるが、いずれのデジタル画像／グラフィックプロセッサも他のデジタル画像／グラフィックプロセッサまたはマスタプロセッサ60にこのコマンドワードを送ることはできない。

【0060】コマンドワードのGビット（ビット13）は、メッセージインタラプトを表示する。いずれのデジタル画像／グラフィックプロセッサもこのメッセージインタラプトを他のデジタル画像／グラフィックプロセッサまたはマスタプロセッサ60に送ることができる。かかるコマンドワードで指定されたデジタル画像／グラフィック

プロセッサは、そのメッセージインタラプトフラグをセットし、メッセージインタラプトがインタラプトイネーブルレジスタINTEN110のビット20によりイネーブルされると、メッセージインタラプトを取り込む。好ましい実施例では、このコマンドワードは転送コントローラ80に送られることはない。

【0061】デジタル画像／グラフィックプロセッサが自身にコマンドワードを発生し、Hビットにより自らを停止させたり、またはIビットによってインストラクションキャッシュをフラッシュさせると、このコマンドワードを実行するには、このコマンドワードは対応するデジタル画像／グラフィックプロセッサの指定ビットセットを有していなければならない。これは一貫性を持たせ、かつコマンドワード機能を将来的に拡張できるようにするためである。

【0062】図5は通信レジスタCOMMのフィールド定義を略図で示す。F、S、QおよびPビット（ビット31～28）は、デジタル画像／グラフィックプロセッサ71、72、73または74および転送コントローラ80からのバケット転送の通信に使用される。FおよびSビットは通常の読み出し／書き込みビットであり、PビットはSビットが0であるか、または同時に0にクリアされる場合に限り書き込みが可能である。9ビットは、読み出し専用であり、バケット転送は転送コントローラ80によるデータ移動のためのデジタル画像／グラフィックプロセッサ71、72、73または74によるリクエストである。これらデータ移動ではマイクロプロセッサ集積回路100の内部のメモリ11～14および21～40だけを必要とする場合もあれば、内部メモリと外部メモリの双方を必要とする場合もある。バケット転送はリンクされたリスト構造として記憶され、各デジタル画像／グラフィックプロセッサに対しては、一度に一つのバケット転送しかアクティブにできない。リクエスト中のデジタル画像／グラフィックプロセッサ71、72、73または74に対応するパラメータメモリ25、30、35または40内の専用アドレスにおけるリンクされたリストポインタは、アクティブなリンクされたリストの開始点をポイントする。リンクされたリストにおける各エントリーは、次のリストエントリーに対するポインタを含む。

【0063】バケット転送の初期化を行うには次のステップが必要である。まずデジタル画像／グラフィックプロセッサは、対応するパラメータメモリに所望するバケット転送パラメータをセットする。次にデジタル画像／グラフィックプロセッサは対応するパラメータメモリに所定のアドレスの16進数0100#0FCにリンクされたリストの第1リンクのアドレスを記憶し、ここで#はデジタル画像／グラフィックプロセッサの番号と置換される。Pビット（ビット28）の1へのセットにより、バケット転送の転送コントローラ80に警告が与え

られる。デジタル画像／グラフィックプロセッサはFビット（ビット31）を1にセットすることにより高い優先度をリクエストしたり、またはFビットをクリアすることにより低い優先度をリクエストできる。

【0064】転送コントローラ80はPビットがセットされたことを認識し、Fビットのステートに基づきパケット転送への優先度を割り当てる。転送コントローラ80はPビットをクリアし、Qビットをセットすることによりパケット転送が待ち行列内にあることを表示する。次に転送コントローラ80は対応するパラメータメモリ内の所定のアドレス、すなわち16進数の0100#0FCにアクセスし、リンクされたリストに基づくパケット転送のサービスを行う。パケット転送が完了すると、転送コントローラ80はQビットを0にクリアし、待ち行列がもはやアクティブでないことを表示する。デジタル画像／グラフィックプロセッサは、パケット転送が完了したかどうかを表示するため、このビットを周期的に読み出すことができる。これとは別に、パケット転送自体がパケット転送終了時にリクエスト中のデジタル画像／グラフィックプロセッサをインタラプトするように転送コントローラ80に命令することもできる。この場合、転送コントローラ80はビット19すなわちパケット転送エンドインタラプトビットPTENDをインタラプトフラグレジスタINTFLG115にセットすることにより、インタラプトをデジタル画像／グラフィックプロセッサへ送る。パケット転送サービス中に転送コントローラ80がエラーを生じると、ビット18、すなわちパケット転送エラーインタラプトビットPTERRORをインタラプトフラグレジスタINTFLG115にセットすることにより、デジタル画像／グラフィックプロセッサにインタラプト信号を送る。デジタル画像／グラフィックプロセッサは表1および適当なインタラプトサービスルーチンに記載された位置に記憶されている適当なインタラプトベクトルを有する。

【0065】デジタル画像／グラフィックプロセッサは、転送コントローラ80が先のリクエストにサービスしている間、別のパケットをリクエストできる。この場合、デジタル画像／グラフィックプロセッサはQビットが1である間にPビットを1にセットする。これが行われると、転送コントローラ80はインタラプトフラグレジスタINTFLG115のビット17をセットすることにより、デジタル画像／グラフィックプロセッサにパケット転送ビジーインタラプトPTBを送る。転送コントローラ80はPビットを0にクリアする。リクエスト中のデジタル画像／グラフィックプロセッサのインタラプトサービスルーチンは、第1パケット転送が待ち行列内にある間、第2パケット転送を一時中止し、パケット転送をキャンセルしたり、他のある訂正措置をとることができる。このような特徴により、デジタル画像／グラフィックプロセッサは通信レジスタCOMM120のQ

ビットを最初にチェックすることなく、パケット転送を送ることができる。

【0066】デジタル画像／グラフィックプロセッサは、Sビットを1にセットすることによりパケット転送のサービスを一時中止にできる。転送コントローラ80は、Sビットが1であることを検出する。パケット転送が待ち行列内にある間このことが起きると、転送コントローラ80はQビットをPビットにコピーし、Qビットをクリアする。これによりPビットは一般に1にセットされる。リクエスト中のデジタル画像／グラフィックプロセッサ内のソフトウェアは、SおよびPビットのステータスを変えることができる。転送コントローラ80は一時中断されたパケット転送のリンクされたリスト内の位置をメモリに保持する。Sビットが0であり、Pビットが同時に1であると、転送コントローラ80が判断すると、一時中止されていたパケット転送は再開される。

【0067】通信レジスタCOMM120の同期ビットフィールド（ビット15～8）は、同期したマルチインストラクション、マルチデータモードで使用される。このフィールドは、同期したマルチインストラクション、マルチデータモードを可能にするロックインストラクションLCKおよびこのモードをディスエーブルするアンロックインストラクションUNLCKによって境界が定められたインストラクションのために作動する。ビット11～8は、インストラクションのフェッチをデジタル画像／グラフィックプロセッサ74、73、72および71とそれぞれ同期化すべきかどうかを表示している。これらビットのいずれかにおける1は、対応するデジタル画像／グラフィックプロセッサが先のインストラクションの実行を完了したことを表示するまでのデジタル画像／グラフィックプロセッサがインストラクションフェッチを遅延することを示している。このデジタル画像／グラフィックプロセッサを同期すべき他のデジタル画像／グラフィックプロセッサは、同様にして通信レジスタCOMM120内の対応するビットをセットすることにある。デジタル画像／グラフィックプロセッサが同期したマルチインストラクション、マルチデータモードにある場合、それ自身に対応する同期ビットをセットする必要はないが、セットしても害はない。ビット15～12は、8つのデジタル画像／グラフィックプロセッサまで拡張する場合に保留したものである。

【0068】通信レジスタCOMM120のDIGP#フィールド（ビット2～0）は、マルチプロセッサ集積回路100上の各々の特定のデジタル画像／グラフィックプロセッサに対してユニークとなっている。これらビットは読み出し専用であり、これらビットに書き込もうとする試みは不可能である。これは同地つでないデジタル画像／グラフィックプロセッサ71、72、73および74の一部にすぎないからであり、ビット1～0は表3に示すような特定のデジタル画像／グラフィックプロ

セッサを識別する2ビットのコードに配線で発生できる。

【0069】

【表3】

COMM フィールド		パラレル プロセッサ
1	0	
0	0	DIGP0 (71)
0	1	DIGP1 (72)
1	0	DIGP2 (73)
1	1	DIGP3 (74)

【0070】ビット2は8個のデジタル画像／グラフィックプロセッサを有するマルチプロセッサ集積回路100で使用するために保留されていることに留意されたい。現在好ましい実施例は、このビットは、4つのデジタル画像／グラフィックプロセッサ71、72、73および74のすべてに対し0となるように、配線で発生される。

【0071】通信レジスタCOMM120のこの部分は特定のデジタル画像／グラフィックプロセッサを識別するように働き、デジタル画像／グラフィックプロセッサの識別番号は通信レジスタCOMM120と7（16進数0000007）とAND論理演算することによって抽出できる。例えば、インストラクションD0=COMM&7がこれを行う。このインストラクションは通信レジスタCOMM120のビット2～10内のデータだけをリターンさせる。このインストラクションは8個のデジタル画像／グラフィックプロセッサを有する実施例に対して適しており、各デジタル画像／グラフィックプロセッサに対応するデータメモリおよびパラメータメモリのアドレスは、そのデータ画像／グラフィックプロセッサの識別に依存するので、識別番号によりソフトウェアがこれらに対応するメモリのアドレスを計算できる。このような識別番号を使用することにより、プログラムを実行する特定のデジタル画像／グラフィックプロセッサと独立したソフトウェアを書き込むことが可能となる。デジタル画像／グラフィックプロセッサと独立したプログラムが、対応するパラメータメモリのベースアドレスと、データメモリのベースアドレスに対しレジスタPBAおよびDBAを使用することもできる。

【0072】マルチプロセッサ集積回路100は小エンディアンまたは大エンディアンフォーマットで作動できる。特に表記しなければ、図および表は大エンディアンフォーマットでのオペレーションを示す。内部レジスタおよび外部データバスのためのビットナンバリングは、常に右側にビット0を有する小エンディアンの規定に従う。64ビットのワード内のバイトは、小エンディアンモードでは右側からアドレス指定され、大エンディアンモードでは左側からアドレス指定される。常に32ビットオペレーションを用いて内部レジスタにアクセスす

ることにより混乱を受けることができる。他の実施例でバス幅を広くしたり狭くしたりすることもできる。16ビットの整数倍であるバス幅が最も有効であると信じられる。

【0073】転送コントローラ80は、マルチプロセッサ集積回路100のうちのマスタプロセッサ60およびデジタル画像／グラフィックプロセッサ71、72、73、74と外部メモリ、例えばビデオRAM5および6並びにメモリ9との間のインターフェースとなっている。転送コントローラ80は、いくつかの自律的メモリオペレーションのみならずプロセッサによってリクエストされるメモリオペレーションを実行する。これらについては後述する。転送コントローラ80はすべてのメモリ11、12、13、14、15、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39および40にアクセスをするクロスバー50に、64ビットのバス接続をしている。従って転送コントローラ80はインストラクションキャッシュ、データメモリおよびパラメータメモリのすべてにアクセスできる。第2の64ビットのバスは、画像システムバスに接続し、オフチップアクセスを行う。

【0074】転送コントローラ80はそのホールド／ホールドアクノージング機構を使用することにより、画像システムを介する外部メモリへのアクセスを外部ホストが行うことを可能にしている。転送コントローラ80はキャッシュミスおよびパケット転送リクエストを行い、外部DRAM／VRAMのリフレッシュを行い、フレームコントローラ90が必要とするシリアルレジスタ転送サイクルを実行し、VRAMをベースとするディスプレイ／キャプチャバッファを更新する。メモリ22、23、24、27、28、29、32、33、34、37、38および39を用いる転送コントローラ80のメモリオペレーションは、通常、デジタル画像／グラフィックプロセッサ71、72、73、74またはマスタプロセッサ60からのパケット転送リクエストに応答して行われる。パケット転送は、オンチップメモリとオフチップメモリとの間でデータを転送する極めてフレキシブルな方法を提供するものである。転送コントローラ80は直接外部アクセス（DEA）サイクルを用いることにより、プロセッサに直接データを送ることもできる。直接外部アクセスサイクルはデジタル画像／グラフィックプロセッサ71、72、73、74がオフチップメモリにアクセスできるようにすると共に、外部メモリにアクセスする際にマスタプロセッサ60がそのデータキャッシュメモリをバイパスできるようにする。転送コントローラ80は種々のリクエストの優先権を定め、必要であればパケット転送リクエスト間で外部メモリインターフェースをタイムシェアする。マスタプロセッサ60、デジタル画像／グラフィックプロセッサ71、72、7

3、74、フレームコントローラ90およびホスト処理システム1からのリクエストのすべては固定された優先度(順位)決定方法によりサービスされる。同じ優先度の多数のリクエストがペンディングであると、転送コントローラ80はラウンドロビン方法に基づいてこれらにサービスを行う。

【0075】転送コントローラ80はプロセッサからの多くの異なるタイプのリクエストを処理しなければならない。最適なシステムの性能を保つには、これらリクエストは緊急度および重要性によって優先度が定められる。転送コントローラ80はこれらの異なる優先度に基づいて作動するので、クロスバー50に対する自己の優先度はサイクルごとに変わり得る。

【0076】図6は、画像システムバス上での転送コントローラ80のオペレーションの優先度を示す。これは階層200として示されている。同じ優先度の多数のリクエストが受信されると、転送コントローラ80はこれらをラウンドロビン法で処理する。これについては図6に示されている。いずれのプロセッサもパケット転送に対して一つのアクティブな優先度しか有することができない。マスタプロセッサ60が緊急優先度、高い優先度および低い優先度のパケット転送リクエストを送ることができるが、デジタル画像/グラフィックプロセッサ71、72、73、74は、高低優先度のパケット転送に限定されている。

【0077】最高優先度201はホスト処理システム1による外部バスリクエスト(HRRQ)のサービスである。転送コントローラ80はホストリクエストライン上の信号にตอบสนองし、画像システムバスの制御をホスト処理システム1の下に置くことができる。

【0078】次に低い優先度202は、フレームコントローラ90からのメモリリクエストのサービスである。次に低い優先度203は、緊急なダイナミックランダムアクセスメモリ(DRAM)のリフレッシュリクエストのサービスである。更に下に述べるように、DRAMリフレッシュリクエストの所定のバックログが形成された際に、これら緊急DRAMリフレッシュリクエストが生じる。

【0079】次の優先度204は、マスタプロセッサ60のインストラクションキャッシュメモリ11および12、データキャッシュメモリ12および13のサービス並びにインタラプトがディスエーブルされたときのマスタプロセッサ60の緊急パケットリクエストである。各プロセッサに含まれる関連するキャッシュロジックにより、キャッシュに関連する転送コントローラ80のすべてのオペレーションが自動的にリクエストされる。リクエストされたデータは外部メモリ位置からリクエスト中のプロセッサのデータキャッシュメモリ内の適当なサブブロックへ移動される。転送コントローラ80はマスタプロセッサ60が必要とする際に、マスタプロセッサ60

0のためにデータキャッシュメモリ13および14のデータサブブロックもセーブする。緊急パケット転送リクエストはマスタプロセッサ60によってリクエストできるだけであり、パケット転送リクエスト内の特定のビットのセッティングを含む。これについては後述する。マスタプロセッサ60がインタラプトをイネーブルする場合、これらデータ転送は優先度が低いことに留意されたい。

【0080】次に低い優先度は、デジタル画像/グラフィックプロセッサ71、72、73、74またはマスタプロセッサ60のインストラクションキャッシュメモリ11および12、データキャッシュメモリ12および13のインストラクションキャッシュリクエストまたはダイレクト外部アクセス(DEA)リクエストのサービス、並びにインタラプトがイネーブルされている時のマスタプロセッサ60の緊急パケット転送リクエストを行うことである。ここで、同じレベルの同時のアクセスリクエストは、リクエストしているプロセッサに応じてラウンドロビン法により処理されることに留意されたい。同じプロセッサに対するキャッシュサービスリクエストおよびダイレクト外部アクセスリクエストは、次のプロセッサにトークンを送る前にサービスされる。

【0081】次の優先レベル206は、高い優先度のパケット転送である。図6に示すように、種々のプロセッサから生じたパケット転送は、ラウンドロビン法で処理される。マスタプロセッサ60またはデジタル画像/グラフィックプロセッサ71、72、73または74のうちの一つがリクエストされたデータを待っている場合、通常、高い優先度のパケット転送を用いる。転送コントローラ80は次の優先レベル207を有する低い優先度のパケット転送に対して同様なラウンドロビン法を用いる。マスタプロセッサ60またはデジタル画像/グラフィックプロセッサ71、72、73または74のうちの一つが、リクエストされたデータを待っていない場合、このプロセッサは低い優先度のパケット転送を通常用いる。後述するように、リクエスト中のプロセッサはパケット転送を緊急にするか、高い優先度にするか、または低い優先度にするかを表示する。

【0082】各リクエストの優先原理について、下に説明する。外部デバイス、例えばホスト処理システム1は、欲する場合、迅速なアクセスを行うことができない。外部デバイスは後述するように、REQ[1:0]をモニタすることにより、必要であれば高い優先度のリクエストに対し転送コントローラ80に画像システムバスを戻すことができる。フレームコントローラ90のリクエストはビデオディスプレイまたはビデオキャプチャを悪化することなく、タイムクリティカルなVRAM転送サイクルが生じるように、第2の優先度を受け入れる。ホストリクエストサイクルおよびフレームコントローラ90のリクエストは、間欠的にしか発生し

ないので、高い優先度を必要とする緊急DRAMリフレッシュはその下の優先度が与えられる。次の優先度はマスタプロセッサ60のキャッシュサービス、ダイレクト外部アクセスおよびマスタプロセッサ60のインタラプトがディスエーブルされている際に生じる緊急優先度のパケット転送サイクルである。これにより、通常インタラプトをディスエーブルするマスタプロセッサ60のインタラプトサービスルーチンは、システム性能を最大にできるようにできるだけ迅速に実行することができる。デジタル画像／グラフィックプロセッサ71、72、73、74のキャッシュサービスおよびダイレクト外部アクセスリクエストは次の優先度である。リクエストのサービスがなされるまで、プロセッサはアイドル状態であるので、これらを迅速にサービスすることが重要である。マスタプロセッサ60のキャッシュサービス、ダイレクト外部アクセスリクエストおよび緊急パケット転送も、インタラプトがイネーブルされている場合のこの優先度にある。その理由は、マスタプロセッサ60はシステム内の他の場所からのインタラプトリクエストにサービスしていないからである。高い優先度のパケット転送はリクエスト中のプロセッサが転送を終了するのにデータを待っていること、または転送コントローラ80が外部バスバンド幅を最適にするよう、クロスバーアクセスのためにデジタル画像／グラフィックプロセッサ71、72、73および74よりも高い優先度とすることが必要であることを意味している。低い優先度のパケット転送は、プロセッサがデータを待っていないことを意味しているので、これらには極めて低い優先度が与えられる。意図する実施例では、トリックルリフレッシュサイクルに最下位の優先度が与えられる。これらサイクルは、外部バスがアイドル状態であり、リフレッシュバックログが0でない場合に実行されるだけである。これは、バックログを小さくし、後の時間に高い優先度の緊急リフレッシュがリクエストされる可能性を少なくするのに役立っている。

【0083】転送コントローラ80が同じ優先度の異なるプロセッサから多数のリクエストを受けるときはいつも、これらの中でラウンドロビン法を行う。このラウンドロビン法は、固定された周期的な優先方法である。すなわちこのことは、ラウンドロビンからいずれのプロセッサも除くことができず、サイクル内のプロセッサの順序を変えることができないことを意味している。特定のプロセッサのリクエストが完了すると、常にチェーン内でペンディング中のリクエストを有する次のプロセッサにラウンドロビントークンが渡される。他のプロセッサからの等しい優先度のリクエストをサービスする必要があるときは、これによって一つのプロセッサが転送コントローラ80を独占することが防止される。

【0084】転送コントローラ80のサービスを行うリクエストレベルに応じ、転送コントローラ80のクロス

バーの優先度はダイナミックに変わる。これについては、図7において階層210として示されている。転送コントローラ80には次の優先度が割り当てられる。緊急優先度のパケット転送リクエスト、キャッシュサービスリクエスト、ダイレクト外部アクセスリクエストのサービスを行っているとき、またはそのパイプラインをフラッシングしているときに、転送コントローラ80はマスタプロセッサ60の優先度212の上の優先度で作動する。転送コントローラ80が、緊急DRAMリフレッシュリクエスト、フレームコントローラ90からのリクエスト、ホストインターフェースリクエストまたはソフトリセットを受けるときはいつも、そのパイプラインをフラッシュする。これらは完了を待っている待機中の外部サイクルで開始することはできない。パイプラインのフラッシングは、ときどき生じるか、または短期間の間マスタプロセッサ60をロックアウトするだけである。

【0085】転送コントローラ80にはデジタル画像／グラフィックプロセッサ71、72、73、74の優先度214よりも高く、かつ高優先度のパケット転送用マスタプロセッサ60よりも低い優先度213が与えられる。これにより、マスタプロセッサ60をロックアウトすることなく転送コントローラ80に最大の可能な優先度が与えられる。マスタプロセッサ60は一般にコントローラとして使用されるので、長期間これをロックアウトすることは、システムに好ましくない影響が及ぶ。

【0086】転送コントローラ80が低い優先度のパケット転送を行う場合、このコントローラの優先度214はデジタル画像／グラフィックプロセッサ71、72、73、74のラウンドロビンの優先度214よりも低い。これによってパケット転送の優先度が低い際に転送コントローラ80がデジタル画像／グラフィックプロセッサ71、72、73、74からのクロスバンド幅をスチールすることが防止される。デジタル画像／グラフィックプロセッサ71、72、73、74のコーカルメモリアクセスは異なるメモリに向けられ、干渉できないことに留意すべきである。従ってこれらコーカルメモリアクセスはラウンドロビン法で同じステージを有する。

【0087】転送コントローラ80によってより高い優先度のリクエストが受け入れられる場合、このコントローラは新しいリクエストのクロスバーの優先度で現在のオペレーションを完了したり、一時中断したりする。これによりシステム内でのブロックが発生しないよう保証される。従って例えば高い優先度のパケット転送リクエストが受け入れられる場合、高い優先度で低い優先度のパケット転送の一時中断が生じる。

【0088】転送コントローラ80はデジタル画像／グラフィックプロセッサ71、72、73、74のインストラクションキャッシュミス、マスタプロセッサ60のインストラクションおよびデータキャッシュミスに自動的にサービスを行う。多数のキャッシュサービスリクエ

ストが受信されると、転送コントローラ80は図6に示すように、ラウンドロビン法に基づき、これらの優先度を定める。キャッシュミスのサービスが完了すると、転送コントローラ80はリクエスト中のプロセッサにシグナルを送る。マスタプロセッサ60はラウンドロビンでの順番内にサービスされるインストラクションおよびデータキャッシュの双方を有することができる。

【0089】デジタル画像／グラフィックプロセッサ71、72、73、74のインストラクションキャッシュは、各々4つの128バイト（16インストラクション）のサブブロックを含む4つのブロックから成る1方向の組の関連（完全関連）キャッシュである。他の実施例では、他方向の組の関連キャッシュを用いることもできる。デジタル画像／グラフィックプロセッサがキャッシュミスを経験すると、このプログラムフロー制御ユニットは転送コントローラ80に信号を送ってキャッシュミスサービスをリクエストする。デジタル画像／グラフィックプロセッサ71、72、73または74は、インストラクションを入れるキャッシュブロックを決定し、この情報のみならずアドレスも転送コントローラ80へ送る。転送コントローラ80は、外部メモリからの完全サブブロック（128バイト）をフェッチし、これをリクエスト中のデジタル画像／グラフィックプロセッサ71、72、73、74の適当なキャッシュサブブロックへ入れる。次に転送コントローラ80はリクエストのサービスが完了し、プロセッサがそのプログラムの実行を続けることができる旨を、デジタル画像／グラフィックプロセッサ71、72、73または74へ知らせる。

【0090】マスタプロセッサ60は、各々4ブロックから成る4方向の関連する組である。各ブロックは4つの64バイトのサブブロックを含む。マスタプロセッサ60はインストラクションキャッシュまたはデータキャッシュ、または双方に対するサービスをリクエストできる。マスタプロセッサ60のインストラクションキャッシュサービスリクエストは、転送コントローラ80によりフェッチされるサブブロックの大きさがわずかに64バイトであることを除けば、デジタル画像／グラフィックプロセッサ71、72、73、74のキャッシュリクエストと同様に扱われる。

【0091】マスタプロセッサ60のデータキャッシュメモリ13および14は、転送コントローラ80がその内容を外部メモリに書き戻すようにリクエストできる点でインストラクションキャッシュ11および12と異なっている。転送コントローラ80はマスタプロセッサ60のデータキャッシュミスに対し、マスタプロセッサ60のインストラクションキャッシュを用いる場合と同じように、64バイトのサブブロックをフェッチする。しかしながら一致するタグアドレスが発見できず、すべてのブロックを使用したことによりマスタプロセッサ60がブロックミスを起こす場合、このプロセッサはまず最

初に転送コントローラ80にブロックを置換する前に、最後に使用したブロック内にダーティサブブロックを書き戻すことをリクエストする。ダーティサブブロックの書き戻しは、ラウンドロビン優先度内の単一マスタプロセッサ60の順番内で行うことができる。転送コントローラ80は特殊キャッシュインストラクションに 응답してダーティサブブロックを書き戻すようにもリクエストできる。

【0092】転送コントローラ80はマスタプロセッサ60およびデジタル画像／グラフィックプロセッサ71、72、73、74からのすべてのダイレクト外部アクセス（DEA）リクエストを処理する責任を負っている。ダイレクト外部アクセスサイクルはデジタル画像／グラフィックプロセッサ71、72、73、74が外部メモリ内のデータに直接アクセスできるようにすると共に、マスタプロセッサ60がそのデータキャッシュをバイパスできるようにする。ダイレクト外部アクセスには高い優先度が与えられているので、これらアクセスは長さが1バイト、半ワード（16ビット）、1ワード（32ビット）、またはダブルワード（64ビット）となり得る単一アクセスに限られる。これにより、一つのプロセッサが多数のダイレクト外部アクセスサイクルで外部バスを独占することが防止され、これによりダイレクト外部アクセスリクエストおよび他のプロセッサのキャッシュミスのサービスが行われるのが防止される。単一のオフチップメモリ位置への高速アクセス、例えばプログラム可変またはオフチップレジスタが必要なときに、ダイレクト外部アクセスサイクルが使用される。

【0093】デジタル画像／グラフィックプロセッサ71、72、73、74は、それらのパラメータメモリ25、30、35および40、並びにそれらのデータメモリ22、23、24、26、28、29、32、33、34、37、38および39に通常アクセスする。16進数02000000以上のアドレスへのアクセスにより、ダイレクト外部アクセスリクエストは自動的に転送コントローラ80へ送られる。このリクエストはキャッシュダイレクト外部アクセスラウンドロビンにおけるリクエスト中のデジタル画像／グラフィックプロセッサ71、72、73、74の順番に達したときにサービスを受ける。キャッシュミスリクエストおよびダイレクト外部アクセスリクエストの双方は、継続中の場合に1回でサービスを行うことができる。クロスバーを介してアクセスできないオンチップメモリ領域、例えばマスタプロセッサ60のパラメータメモリ16へのデジタル画像／グラフィックプロセッサ71、72、73、74のアクセスは、転送コントローラ80へのダイレクト外部アクセスリクエストに変換される。しかしながらこのダイレクト外部アクセスはフォールトとなる。フォールトを生じさせるようなデジタル画像／グラフィックプロセッサ71、72、73、74のダイレクト外部アクセスサイ

クルは、フォールトしたデジタル画像／グラフィックプロセッサ71、72、73、74のキャッシュサイクルと同じように取り扱われる。

【0094】マスタプロセッサ60はデジタル画像／グラフィックプロセッサ71、72、73、74と若干異なる態様でダイレクト外部アクセスサイクルを用いる。マスタプロセッサ60は、データキャッシュメモリ13および14により通常外部メモリにアクセスする。マスタプロセッサ60はダイレクト外部アクセスサイクルを利用し、データキャッシュメモリ13および14およびアクセスメモリを直接バイパスする。特殊メモリロードまたは記憶インストラクションを用いることにより、ダイレクト外部アクセスサイクルが明瞭に指定される。マスタプロセッサ60にアクセスできないオンチップアドレス、例えばデジタル画像／グラフィックプロセッサ71、72、73、74のインストラクションキャッシュメモリ21、26、31および36に対するかかる特別ロードまたは記憶インストラクションを、マスタプロセッサ60が実行しようとする場合、このオペレーションはダイレクト外部アクセスリクエストに変換され、次に変換コントローラ80によってフォールトされる。アクセス可能なオンチップメモリ領域、例えばオンチップレジスタまたはデジタル画像／グラフィックプロセッサ71、72、73、74のデータメモリ22、23、24、27、28、29、32、33、34、37、38、39への特別メモリロードまたは記憶オペレーションは、通常のロードまたは記憶オペレーションに変換され、転送コントローラ80を介してダイレクト外部アクセスリクエストは行われない。

【0095】図8は、転送コントローラ80の高度なブロック図を示す。下記に、各主要ブロックを簡単に説明する。転送コントローラ80は内部メモリインターフェース301と、外部メモリインターフェース302と、リクエスト待ち行列化および優先度決定回路303と、

キャッシュ、VRAMおよびリフレッシュコントローラ310と、パケット転送用先入れ先出し(FIFO)バッファ311と、キャッシュバッファ312と、ソースレジスタ321およびソース制御ロジック322から成るソースマシン320と、ソースマルチプレクサおよびアライメントロジック330と、宛て先レジスタ341および宛て先制御ロジック342から成る宛て先マシン340と、宛て先マルチプレクサおよびアライメントロジック302とから成る。

【0096】転送コントローラ80はマルチプロセッサ集積回路100による外部メモリへのすべてのアクセスに対して責任を負っている。外部メモリインターフェースは多数の異なるタイプや大きさのメモリデバイスおよび周辺機器に合わせるように設計されている。アクセスされるメモリのタイプは、ダイナミックに決定されるので、各メモリサイクルのタイミングはアクセスされるデバイスに対して最適にすることができる。

【0097】次は、外部メモリインターフェース302を介して外部データアクセスのために使用されるマルチプロセッサ集積回路100の信号のリストである。

【0098】アドレスバスA〔31:0〕。これはマルチプロセッサ集積回路100からの32ビットのバイトアドレスを外部メモリに送るためのものである。このアドレスはDRAMアクセスに対して多重化できる。

【0099】アクセスシフト選択AS〔2:0〕。これらの入力値は転送プロセッサ80による別アドレスへ与えられるシフト量を決定する。表4aおよび4bに示すように、0を含む8つのシフト量がサポートされている。表4aはアドレスバスビットA〔31:16〕上の出力を示し、表4bはアドレスバスビットA〔15:0〕上の出力を示す。

【0100】

【表4】

AS[2:0]	A[31:16]															
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0 0 0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0 0 1	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8
0 1 0	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7
0 1 1	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6
1 0 0	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5
1 0 1	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4
1 1 0	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3
1 1 1	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2

表 4 a

AS[2:0]	A[15:0]															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0 0 0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0 0 1	7	6	5	4	3	2	1	0	X	X	X	X	X	X	2	1
0 1 0	6	5	4	3	2	1	0	X	X	X	X	X	X	X	2	1
0 1 1	5	4	3	2	1	0	X	X	X	X	X	X	X	X	2	1
1 0 0	4	3	2	1	0	X	X	X	X	X	X	X	X	X	2	1
1 0 1	3	2	1	0	X	X	X	X	X	X	X	X	X	X	2	1
1 1 0	2	1	0	X	X	X	X	X	X	X	X	X	X	X	2	1
1 1 1	1	0	X	X	X	X	X	X	X	X	X	X	X	X	2	1

表 4 b

【0101】行アドレス時間の間、アドレスバスA〔31:0〕は、通常のアドレス値を出力する。列アドレス時間の間、アドレスバスA〔31:0〕は表4aおよび4bに示すようにアドレスシフト選択AS〔2:0〕に従ってシフトされたアドレス値を出力する。Xと表示されたアドレスライン出力は、ドライブされず、高インピーダンス状態にセットされたものであることに留意されたい。

【0102】バスサイズ選択BS〔1:0〕。これは64ビット幅よりも狭いデータバスに対するダイナミックなバスの大きさの決定を可能とする。下記の表5に、これら入力信号のコード化を示す。

【0103】

【表5】

*

BS[1:0]	ビットサイズ
0 0	8 ビット
0 1	16 ビット
1 0	32 ビット
1 1	64 ビット

【0104】出力ブロックCLKOUT。このクロック出力は外部ロジックがマルチプロセッサ集積回路100の作動に同期できるようにするものである。別タイミング選択CT〔1:0〕。これらの入力信号は転送コントロール80によって与えられる現在のメモリスサイクルのどのタイミングであるかを決定する。下記の表6にこれら入力信号のコード化を示す。

【0105】

【表6】

*

CT[1:0]	メモリタイミング
0 0	パイプライン状の1サイクル/列
0 1	非パイプライン状の1サイクル/列
1 0	非パイプライン状の2サイクル/列
1 1	非パイプライン状の3サイクル/列

表 6

【0106】列アドレスストローブCAS'〔7:0〕。これら出力は、DRAM/VRAMの反転CASをドライブする。個々のバイトアクセスを可能にするように、8個のストローブが達成される。エンディアンモードにかかわらず、反転CAS〔0〕は、D〔7:0〕でのデータ転送に対応し、CAS'〔1〕は、D〔15:9〕上の転送に対応する。トランスベアレンシーではこれら信号はバイト書き込みストローブとして使用される。データバスD〔63:0〕。この入出力バスは、マルチプロセッサ集積回路100の内外にメモリスサイクル

ル当たり64ビットまでのアクセスを可能とする。

【0107】データバッファ出力イネーブルDBEN'。この出力はデータトランシーバをオンにするのに用いることができる。データ方向インディケータDDIN'。この出力は、データトランシーバに対する方向インディケータとなる。特別機能ピンDSF1およびDSF2。これら出力は特別VRAM機能を選択するのに使用される。フォールト反転FAULT。この入力メモリフォールトが生じたことを、マルチプロセッサ集積回路100に知らせる。

【0108】ページサイズ選択PS〔2:0〕。これら入力は今現在アクセスされているメモリのページ選択をマルチプロセッサ集積回路100に示す。これら入力の使用については、後に更に詳細に説明する。行アドレスストローブ反転RAS。これら出力はDRAM/VRAMの反転RAS入力をドライブする。

【0109】READY。この入力は完了すべきメモリサイクルに対して外部デバイスがレディ状態となっていることを表示する。これはメモリサイクル内に待機ステートを挿入するのに、転送コントローラ80によって使用される。

【0110】リトライRETRY'。この入力メモリがビジーとなっており、転送コントローラ80が再びメモリサイクルを開始すべきことをマルチプロセッサ集積回路100に表示するものである。行ラッチRL'。こ*

*の出力はアドレスバス上に有効な32ビットアドレスが存在していることを表示するものである。

【0111】ステータスコードSTATUS〔4:0〕。この出力は現在の転送コントローラのメモリサイクルのタイプと、起点についての詳細な説明を与える。表7に示すように、行アクセスの開始から列アクセスの開始までに、STATUS〔4:1〕の出力がコード化される。下記のUTIME'と共にこれらをコーディングすることにより、マルチプロセッサ集積回路100のタイミングに発生するメモリタイミングを外部デバイスが発生することができるようになっている。

【0112】

【表7】

STATUS〔4:0〕	アクティビティ
00000	通常の読み出し
00001	通常の書き込み
00010	リフレッシュ
00011	保留
00100	周辺デバイスバケット転送読み出し
00101	周辺デバイスバケット転送書き込み
00110	保留
00111	保留
01000	保留
01001	ブロック書き込み
01010	保留
01100	保留
01101	ロードカラーレジスタ
01110	保留
01111	保留
10000	フレーム0フルシフトレジスタ読み出し
10001	フレーム0フルシフトレジスタ書き込み
10010	フレーム0スプリットシフトレジスタ読み出し
10011	フレーム0スプリットシフトレジスタ書き込み
10100	フレーム1フルシフトレジスタ読み出し
10101	フレーム1フルシフトレジスタ書き込み
10110	フレーム1スプリットシフトレジスタ読み出し
10111	フレーム1スプリットシフトレジスタ書き込み
11000	保留
11001	保留
11010	保留
11011	保留
11100	バケット転送フルシフトレジスタ読み出し
11101	バケット転送フルシフトレジスタ書き込み
11110	保留
11111	保留

表7

【0113】行時間ステータスコードの詳細は次のとおりである。バケット転送、キャッシュミスまたはダイレクト外部アクセスリクエストに対しては通常の読み出しコードが出力される。バケット転送、データキャッシュ書き戻しまたはダイレクト外部アクセスリクエストによって発生される通常の書き込みサイクルに対しては通常の書き込みコードが出力される。トリックルリフレッシュサイクルおよびリフレッシュコントローラからの緊急リフレッシュリクエストにより発生されるバーストリフレッシュサイクルの間にリフレッシュコードが出力される。

【0114】下記に周辺デバイスバケット読み出しおよび書き込みについて更に説明する。周辺デバイスのバケ

ット転送の結果として生じるメモリ読み出しサイクルに対しては、周辺デバイスバケット転送読み出しコードが出力される。このことは、後の列アクセス上のメモリから読み出されるデータを、転送を開始させた周辺デバイスによってラッチすべきことを示している。周辺デバイスのバケット転送の結果として生じるメモリ書き込みサイクルに対しては、周辺デバイスバケット転送書き込みコードが出力される。このことは、周辺デバイスがメモリ内に入れるべきデータでバスをドライブできるように、後の列アクセスの間にマルチプロセッサ集積回路100がデータバスを高インピーダンスにすることを意味している。

【0115】ブロック書き込みは特別なVRAMサイク

ルである。ブロック書き込みアクセスモードでパケット転送によって発生されたVRAMに対し、ブロック書き込みサイクル中にブロック書き込みコードが出力される。アクセスモードとしてブロック書き込みを指定したパケット転送のカラーレジスタロード部分の間でロードカラーレジスタコードが出力される。このサイクルはシステムVRAMのカラーレジスタにデータを入れるのに使用され、このカラーレジスタデータは部分書き込みサイクル中にメモリに書き込まれる。

【0116】シリアルレジスタ転送アクセスモードを使用してパケット転送のソースサイクルの間にパケット転送シフトレジスタ読み出しコードが出力される。このサイクルはシステムのVRAM上で読み出しメモリからレジスタへの転送を行うものである。シリアルレジスタ転送アクセスモードを指定したパケット転送の宛て先サイクルの間に出力される。このサイクルシステムVRAM上での書き込みレジスタからメモリへの転送を実行する。

【0117】このフレームサイクルはフレームコントローラ90によって必要とされるメモリアクセスである。シフトレジスタの読み出しおよび書き込みは特別シリアルレジスタのVRAMサイクルとなっている。位置読み出し転送コードのうちのフレーム0は、フレームコントローラ90によってリクエストされるそれぞれのフル読み出し転送サイクルの間に出力される。これらサイクルはシステムVRAM上でのフル読み出し転送サイクルを実行し、STATUS〔2〕信号はフレーム0のリクエストサイクルでは0であり、フレーム1のリクエストサイクルでは1である。フレームコントローラ90によってリクエストされるフル書き込み転送サイクルの間にフ

レーム0または1の書き込み転送コードが出力される。これらサイクルは、システムVRAM上でのフル書き込み転送サイクルを実行する。STATUS〔2〕信号はフレーム0のリクエストサイクルに対しては0であり、フレーム1のリクエストサイクルに対しては1である。フレームコントローラ90はスプリット読み出し転送サイクルをリクエストするとフレーム0または1のスプリット読み出し転送コードが出力される。これらサイクルはVRAMメモリの行からシリアルレジスタの半分への転送を行う。STATUS〔2〕信号はフレーム0のリクエストサイクルに対しては0であり、フレーム1のリクエストサイクルに対しては1である。フレームコントローラ90によってリクエストされるスプリット書き込み転送サイクルの間に、フレーム0または1のスプリット書き込み転送コードが出力される。これらサイクルはVRAMシリアルレジスタの半分からメモリアレイ内へ転送を行う。STATUS〔2〕信号はフレーム0のリクエストサイクルに対しては0であり、フレーム1のリクエストサイクルに対しては1である。

【0118】別アクセスからのスタートから行アクセスのスタートまでに、表8に示すように、STATUS〔4:1〕の出力がコード化される。この情報は、転送コントローラ80を介してマルチプロセッサ集積回路100により出力され、メモリサイクルをリクエストするモジュールを示す。これら信号は、システム解析およびデバッグのため発生されるが、外部メモリシステムには不要である。

【0119】

【表8】

STATUS [4:0]

アクティビティ

00000	DIGP0	低優先度バケット転送
00001	DIGP0	高優先度バケット転送
00010	DIGP0	キャッシュ
00011	DIGP0	DEA
00100	DIGP1	低優先度バケット転送
00101	DIGP1	高優先度バケット転送
00110	DIGP1	キャッシュ
00111	DIGP1	DEA
01000	DIGP2	低優先度バケット転送
01001	DIGP2	高優先度バケット転送
01010	DIGP2	キャッシュ
01011	DIGP2	DEA
01100	DIGP3	低優先度バケット転送
01101	DIGP3	高優先度バケット転送
01110	DIGP3	キャッシュ
01111	DIGP3	DEA
10000	MP	低優先度バケット転送
10001	MP	高優先度バケット転送
10010	MP	緊急バケット転送インタラプトイネーブル
10011	MP	緊急バケット転送インタラプトディスエーブル
10100		保留
10101		保留
10110	MP	インストラクションキャッシュインタラプトイネーブル
10111	MP	インストラクションキャッシュインタラプトディスエーブル
11000	MP	DEAインタラプトイネーブル
11001	MP	DEAインタラプトディスエーブル
11010	MP	データキャッシュインタラプトイネーブル
11011	MP	データキャッシュインタラプトディスエーブル
11100		フレームメモリ0
11101		フレームメモリ1
11110		リフレッシュ
11111		保留

表8

【0120】転送／出力イネーブル反転TR。転送コントローラからのこの出力は、DRAM出力ドライバおよびVRAMシフトレジスタ転送サイクルをイネーブルする。ユーザータイミング選択反転UTIME。この入力には転送コントローラが反転RASおよび反転CAS

【7:0】のタイミングを変え、ユーザーが自らのメモリタイミングを発生できるようにするものである。この入力にはマルチプロセッサ集積回路100が作動するエンディアンモードを決定するようにリセットでも使用される。書き込みイネーブル反転WE。これは通常は書き込みまたは書き込み転送サイクルが生じていることをメモリに表示する出力である。内部メモリテストモードでは、この信号は内部メモリへの書き込みを行うよう、外部デバイスによってドライブされる入力となっている。

【0121】マルチプロセッサ集積回路100により発生される各外部メモリサイクルは、ページモードサイクルを除く期間中の少なくとも5つのマシンステートである。マシンステートは1クロック期間の長さであり、CLKOUTの降下エッジで始まる。各メモリサイクルは2つの部分、すなわちアドレスサブサイクルとデータサブサイクルとを有する。ページモードサイクルは、アクセスが一つのアドレスサブサイクルとマルチデータサブサイクルを有するこのフォームの拡張である。

【0122】アドレスサブサイクルは外部メモリサイクルの第1マシンステートで開始し、少なくとも4つのマ

シンステート長さである。このときにアクセスのためのアドレスおよびステータスコードが出力される。この時間にDRAMおよびVRAMのための行アドレスがラッチされるので、このサイクル部分は、行アドレス時間とも称される。

【0123】アドレスバスA[31:0]は、現在アクセス中の64ビットワードの開始バイトをポイントする32ビットのアドレスを出力する。このアクセスは転送すべきスタートバイトおよびデータ量に依りて1~8バイトのいずれでもよい。このアドレスはマルチプロセッサ集積回路100の外部メモリスペースをデコードするのに使用される。次に外部デコードロジックは多数の信号をマルチプロセッサ集積回路100へ戻し、アドレスシフト選択AS[2:0]によりアクセスされているデバイスのタイプ、列タイミング選択CT[1:0]により速度、ページサイズ選択PS[2:0]によりページサイズ、バスサイズ選択BS[1:0]によりデータバス幅を表示する。この情報はアドレスサブサイクルの長さのみならず、データサブサイクルの長さ、アドレス指定およびその数を決定するのに使用される。アドレスおよびステータスはRL' またはRAS' によりラッチできる。

【0124】アドレスサブサイクルは、実行されるアクセスのタイプが必要とするような4つのマシンステートを越えるマシンステートの整数だけ自動的に延長され

る。このサブサイクルは更に待機状態の挿入によっても延長される。このサブサイクルは少なくとも1つのマシンステートの長さであり、アドレスサブサイクルの直後に続く。この時には、DRAMおよびVRAMのための別アドレスが出力され、マルチプロセッサ集積回路100を外部メモリとの間でデータが転送される。このメモリサイクル部分は、列アドレス時間と称される。

【0125】データバスD〔63:0〕は、マルチプロセッサ集積回路100と外部メモリの間でデータを転送する。これらデータは書き込みサイクルでドライブされるか、または読み出しサイクルでラッチされるかのいずれかである。バス上の有効データの位置はマルチプロセッサ集積回路100のエンディアンモード、転送データ量およびメモリ幅により決定される。

【0126】この時間の間に出力される列アドレスは、32ビットバイトのアドレスのシフトされた変形例である。アドレスバスA〔31:0〕バス上のアドレスのアライメントはアドレスサブサイクル中にマルチプロセッサ集積回路100へ入力されるアドレスシフト選択AS〔2:0〕によって決定される。

【0127】データサブサイクルの長さは、通常、アドレスサブサイクル中にマルチプロセッサ集積回路100へ入力される列タイミング選択CT〔1:0〕により決定されるような1つ、2つまたは3つのマシンステートである。より長いアクセス時間を必要とするデバイスは、アドレスまたはデータサブサイクルのいずれかに待機状態を挿入できる。

【0128】現在のメモリアクセスは、行時間におけるページサイズ選択PS〔2:0〕入力に基づく先のアクセスと同一方向で、かつ同一メモリページ内にあるときはいつも、転送コントローラ80はページモードサイクルを使用する。ページモードサイクルは後にマルチデータサイクルが続く一つのアドレスサブサイクルから成り、データは同じメモリページ内に限り隣接している必要はない。

【0129】アドレスおよびデータサブサイクルの双方の間で、マルチプロセッサ集積回路100はSTATUS〔4:0〕ピン上にステータスコードを出力する。これらステータスコードは実行中の外部サイクルに関する情報を与えるものである。アドレスサブサイクルの行時間の間、STATUS〔4:0〕ピンは表7に示すようなコードを出力し、実行中のサイクルのタイプを表示する。その行時間ステータスコードは反転RLまたは反転RAS信号によってラッチでき、メモリバンクデコーディングを行ったり、特別ハードウェア機能をイネーブリングするように外部ロジックによって使用される。データサブサイクルの列時間の間、STATUS〔4:0〕ピンに関する情報が変化し、そのサイクルおよびそのリクエスト中のプロセッサについての詳細が与えられる。表8にはこれらアクティビティコードが示されている。これ

らコードはサイクル自体のタイプに関する情報を与えるものでないのに、システムのデバッグを容易にするように、主として使用されるものである。

【0130】転送コントローラ80が外部メモリと正しく通信できるようにするため、アクセス中のメモリがどのタイプであるかを知る必要がある。これはマルチプロセッサ集積回路100のアドレスシフト選択AS〔2:0〕、バスサイズ選択BS〔1:0〕、列時間選択CT〔2:0〕およびページサイズ選択PS〔2:0〕入力を用いることにより、行時間でメモリタイプを識別することにより行われる。マルチプロセッサ集積回路100は外部メモリアドレスおよび行時間ステータスコードを出力し、これら入力をサンプリングし、メモリタイプを決定する。これによりアドレスをデコードし、メモリ識別信号を適当なレベルにドライブするための外部ロジック時間が与えられる。選択されたメモリタイプは、次のアドレスサブサイクルまで有効な状態に留まる。

【0131】マルチプロセッサ集積回路100はDRAMのみならずSRAMもサポートしているので、アドレスバス上に多重化された行および列アドレスを発生しなければならない。マルチプロセッサ集積回路100は常に行時間にフルの32ビットバイトのアドレスを出力する。列時間で、この回路はバス上のアドレスをシフトしてDRAMによってすでにラッチされている行アドレスに列アドレスを一致させなければならない。アレイサイズはデバイスの行/列アドレスビットの数を決定するので、列アドレスを正しく合わせるにはマルチプロセッサ集積回路100はアクセスするDRAMのアレイサイズに知っている必要がある。これはアドレスシフト選択AS〔2:0〕入力を使用して行われる選択である。外部論理は行時間においてマルチプロセッサ集積回路100によって出力されるアドレスをデコードし、アドレスシフト選択AS〔2:0〕入力に3ビットのシフトコードを供給する。転送コントローラ80はこの値をサンプリングし、ラッチし、これを使ってダイナミックメモリに必要なようにアドレスを多重化するか、またはスタティックメモリおよび周辺機器と使用するため、多重化しなしておくかを決定する。

【0132】表4aおよび4bは、サンプルされたアドレスシフト選択AS〔2:0〕の値が列時間に出力されるアドレスにどのように影響するかを示している。値が000である場合、各々の後の列アドレスにたいしてシフトされていない32ビットのアドレスが出力される。値が0でない場合、表4aおよび4bが示すように、その後の別アドレスがシフトされる。シフト値は8~14ビットの範囲であるが、このことは64K×N個~256M×N個のアレイサイズに対応する8~14個のアドレスピンを備えたダイナミックメモリがサポートされることを意味している。アドレスバスA〔2:0〕はシフト量に関係なくバイトアドレスビット0~2を常に出力

する。これはダイナミックバスのサイジングをサポートするため行われる。

【0133】例えば1M×4個のDRAMが64ビットワイドのコンフィギュレーションでデータバスに接続されていると仮定する。これらメモリは各々10ビットの行および列アドレスを必要とする。マルチプロセッサ集積回路100のアドレスシフト選択AS〔2:0〕はバイトアドレスを示しており、メモリバンクは64ビット幅であり、ここのバイトは反転CAS〔7:0〕ストローブにより制御されるので無視できる。このことは、メモリはビットA〔3〕で始まる20個の隣接するアドレスビットを必要とする。表4aおよび4bを検討すると、011アドレスシフト選択のAS〔1:0〕の値は16ビットの多重化されたアドレスを与えることが判る。ビットA〔3〕の開始アドレスが列時間におけるアドレスピンA〔13〕に対応しているので、A〔13〕で開始する10個のマルチプロセッサ集積回路100のアドレスピンにはDRAMアドレスピンが接続されている。

【0134】可能であれば転送コントローラ80はページモードサイクルを実行するので、コントローラは現在アクセス中のメモリのためのページまたは行境界を、いつクロスして新しいページ上の行のアクセスを行うことができるかを知る必要がある。現在のアクセスに対するページサイズは、外部ロジックによりページサイズ選択PS〔2:0〕ピンに置かれる3ビットの値により表示される。転送コントローラ80は行時間でこれらピンをサンプリングし、これらを使用して変化しているどのアドレスピンがページの変化を表示しているかを決定する。サンプリングされた値は、次の行アクセスまでに転送コントローラ80によって保持される。ページサイズ選択PS〔2:0〕に表示されたページサイズは、必ずしもアドレスシフト選択AS〔2:0〕上に表示されたシフト量に対応しているわけではないが、この理由は、多数のバンクをインターリーブすることができるからである。

【0135】外部メモリアccessが起きるといつも、転送コントローラ80はアドレスの21個の最高位ビット*

*を内部LASTPAGEレジスタ360に記録する。各々のその後の列アクセスのアドレスは、この値と比較される。これは図9にプログラムされている。ページサイズ選択PS〔2:0〕に入力された値は、比較中にLASTPAGEレジスタ360の6個の最小位ビットを選択的に無視するのに使用される。この比較において、ページサイズ選択PS〔2:0〕-1の値に等しいビット番号よりも低いLASTPAGEレジスタは無視される。常に15個の最高位ビットが比較され、常に次のアドレスの11の最小位ビットが無視される。LASTPAGEレジスタ136のイネーブルされたビットと次のメモリアドレスとが一致しない場合、ページが変わり、行アドレスサイクルと共に次のメモリアドレスが始まる。ページサイズ選択PS〔2:0〕=00であれば、ページモードがディスエーブルされ、別の行アクセスと共にその後のサイクルが開始される。

【0136】例えばアクセスされているメモリが他のバンクとインターリーブされない64ビットのデータバスとして接続されている16個の1M×4個のDRAMから成ると仮定する。各メモリデバイスは2の10乗ビットの行サイズを有する。DRAMのページモードサイクルの間に、単一行内の任意の位置にアクセスできるので、コンフィギュレーションのためのページサイズは2の10乗個の位置/ページ×8バイト/位置=8Kバイト/ページとなる。図9が示すように、これは011のページサイズ選択PS〔2:0〕に対応する。転送コントローラ80が行時間でページサイズ選択PS〔2:0〕=011をサンプリングすると、コントローラはページ境界がクロスされたかどうかを判断するため、その後のアクセスのビット31~13をチェックする。これはこのような特殊コンフィギュレーションに対するDRAMの行アドレスビットおよびバンクデコードビットに対応する。図9に、8個の可能なページサイズ選択PS〔2:0〕のための比較されたアドレスビットおよびページサイズが示されている。

【0137】

【表9】

PS〔2:0〕			ページサイズ	比較されるビット
0	0	0	no page mode	31-0
0	0	1	2 K	31-11
0	1	0	4 K	31-12
0	1	1	8 K	31-13
1	0	0	16 K	31-14
1	0	1	32 K	31-15
1	1	0	64 K	31-16
1	1	1	128 K	31-17

【0138】LASTPAGEレジスタ360は関連する有効ビット361を有する。この有効ビット361は、LASTPAGEレジスタに記憶されたデータが有効なものとして処理されたか、または無効なものとして

みなされたかを示す。LASTPAGEレジスタ360はリセット、ホストアクセス、フォールトおよび再試行の後、またはアクセス方向の切り替え、例えば読み出しと書き込みとの切り替えの後に、有効なもののみなされ

る。これにより、その後のアクセス前に常に行アクセスが強制的に行われる。更にLASTPAGEレジスタ360はリフレッシュサイクル、フレームコントローラ90のリクエストしたサイクルおよびパケット転送の発生したシリアルレジスタ転送サイクルの前後で無効とみなされるので、これらサイクルは常にシングルのノンページモードサイクルとして生じる。周辺デバイスのパケット転送の特殊な例では、周辺デバイス転送ステータスコードが出力されるように、常に行アクセスで転送が開始する。新しいステータスコードが出力できるように、転送の終了時にLASTPAGEレジスタ360のデータは無効とみなされる。しかしながら転送中はLASTPAGEレジスタ360は周辺デバイス転送が可能な限りページモードサイクルを利用するように、通常作動する。

【0139】現在のアクセスのためのバスサイズを決定するように、行時間でバスサイズ選択PS[1:0]ピンがサンプリングされる。転送コントローラ80は表5に示すように、8、16、32または64ビットのバスサイズをサポートする。バスサイズをセットすることにより、各列アクセス中に転送コントローラ80が転送できるバイトの最大数が決定される。リクエストされたバイト数がバスサイズを越える場合、転送コントローラ80は転送を完了するように自動的にマルチアクセスを実行する。選択されたバスサイズは、転送にデータバスのどの部分が必要となるかも決定する。64ビットのメモリに対しては、全バスを利用できる。32ビットメモリに対しては、ビッグエンディアンモードでデータバスD[63:32]ピンを使用し、リトルエンディアンモードではデータバスD[31:0]ピンを用いる。ビッグエンディアンモードおよびリトルエンディアンモードではそれぞれ16ビットバスはデータバスD[63:48]ピンおよびデータバスD[15:0]ピンを利用し、8ビットバスはデータバスD[63:56]ピンおよびデータバスD[7:0]ピンを用いる。どのようなバスサイズを使用するにせよ、転送コントローラ80は常にデータをバスの適当な部分に一致させ、適当な反転CASストロブを附勢し、有効なバイトしか転送されないように保証する。

【0140】転送コントローラ80はメモリタイミングの4つの基本的組をサポートしている。これにより待機状態を使用することなく、1列アクセス当たり1、2または3クロックサイクルを選択することができる。すべてのタイプは性質がDRAMに類似しているが、各々は特定タイプのDRAMまたはSRAMに適している。使用するタイミングは表6に示すように、列タイミング選択CT[1:0]入力によって決定される。現在アクセス中のメモリタイプを決定するため、アドレスをデコードした後外部デコードロジックはこれらピンを適当なレベルまでドライブする。パイプライン化されて

いない1サイクル/列タイミングは、極めて高速のアクセス時間を有するデバイスと共に使用されるよう設計されている。パイプライン化された1サイクル/列タイミングは、アクセスがパイプライン化されていることを除けば1サイクル/列タイミングに類似している。1サイクルで1つのアドレスを出力し、次のサイクルでそのアドレスに対応するデータがアクセスされる。このタイミングはパイプライン化されたページモードサイクルをサポートするDRAM/VRAMと共に使用するものであるが、同期式SRAMデバイスと共に使用することもできる。パイプライン化されていない2サイクル/列タイミングサイクルは、SRAMおよび高速DRAMデバイスに対して列アクセス時間の2クロックサイクルを発生する。更にこれらタイミングサイクルは、行アドレス時間に1サイクルを加えることも行う。パイプライン化されていない3サイクル/列サイクルは、DRAMおよびその他の低速デバイスに用いるためのものである。このサイクルは列アクセス時間の3クロックサイクルを発生する外に、行アクセス時間に2つのクロックサイクルを加える。更に選択された列タイミングは全ページの間、すなわち次の行アクセスが行われるまで有効なままである。外部デバイスの役割は、発生されたアドレスに基づき、適当な列タイミング選択CT[1:0]を供給することである。

【0141】ブロック書き込みサイクルおよびロードカラーレジスタサイクルの間、バスサイズ選択BS[1:0]ピンは異なる目的に対して使用される。ブロック書き込みは64ビットバスに対してしかサポートされていないので、バスサイズ情報は不要である。その代わりに、アドレス指定されたメモリがサポートするブロック書き込みのタイプを表示するように、バスサイズ選択BS[1:0]が使用される。表23には、これらサイクルの間のBS[1:0]の値が示されている。ブロック書き込みについては後に詳細に説明する。

【0142】図8に示した意図する実施例では、転送コントローラ80の外部メモリインターフェース302によりマルチプロセッサ集積回路100の外部メモリサイクルが発生される。外部メモリインターフェース302はメモリインターフェース信号の変化を制御するよう、一連のステートを発生する複雑なステートマシンを含んでいる。発生されるステートおよびそれらのシーケンスは、実行されるサイクルのタイプ、アクセスされるメモリの列タイミング、次に実行すべきアクセスおよびフォールト等のような外部または内部事象に基づき変わる。意図する実施例では、マシンステートを含むが、当業者であればリードオンリーメモリに固定プログラムを有するマイクロコントローラまたはプログラマブルマイクロプロセッサによって、これら機能を実行できることは理解できよう。

【0143】図10は、外部メモリインターフェース3

02のための方法を定めた完全状態図を示す。多数のステートとステート遷移が存在するが、これらのシーケンスは基本的には、実行されるメモリアクセスに対して選択される列タイミングに依存する。更にステートは2つのグループ、すなわち行時間ステートと列時間ステートとに分解できる。

【0144】行時間ステートは、各メモリアクセスのアドレスサブサイクルすなわち行時間を構成し、外部メモリーインターフェース302は新しいページアクセスを開始する時はいつもこれらステートに入る。アドレスされるメモリアドレスが決定されるのは、これらステートの間である。各行アクセスの間には、最低4つの行ステートが生じる。これらは次のように定義される。

【0145】すべてのメモリアクセスに対する開始ステートはr1ステート370である。ステート370中、転送コントローラ80は行アドレスA[31:0]およびサイクルタイプSTATUS[4:0]を出力し、すべての制御信号を不動作ステートにドライブする。

【0146】すべてのメモリアクセスに共通な次のステートは、r2ステート371である。ステート371中では、転送コントローラ80は反転RLハイをアサートし、データ転送方向に従ってDDIN'をドライブする。ステート371中に、転送コントローラ80はアドレスシフト選択AS[2:0]、バスサイズ選択[1:0]、列タイミング選択CT[1:0]、ページサイズ選択PS[2:0]および反転UTIME入力をサンプリングする。

【0147】外部メモリーインターフェース302は、ブロック書き込みおよびロード色レジスタを含む読み出しおよび書き込みのために、r3ステート372となる。ステート373中、転送コントローラ80はDBEN'、DSF1、DSF2、反転TRGおよびWE'をアクティブ行時間レベルにドライブし、FAULT'、READYおよびRETRY'入力をサンプリングする。

【0148】外部メモリーインターフェース302は、リフレッシュの間に限りrf3ステート373となるステート373中、転送コントローラ80は、すべてのCAS'[7:0]ストロブを附勢し、FAULT'、READYおよびRETRY'入力をサンプリングする。

【0149】外部メモリーインターフェース302は、シフトレジスタ転送サイクルに限り、tr3ステート374となる。ステート374は外部的にはr3ステート372と同一である。

【0150】外部メモリーインターフェース302は、列タイミング選択CT[1:0]=11のときに、3サイクル/列アクセスの間に限り、r4ステート375となる。信号遷移は生ぜず、転送コントローラ80はRETRY'入力をサンプリングする。

【0151】列タイミング選択CT[1]=1のときに、r5ステート376は2および3サイクル/列アクセスと共通である。ステート376中、転送コントローラ80は、RAS'出力を低レベルにドライブし、RETRY'入力をサンプリングする。

【0152】外部メモリーインターフェース302はリフレッシュを除くすべてのアクセスの間でrf6ステート377となる。ステート377中、転送コントローラ80はRAS'が既に低レベルでなければこれを低レベルにドライブし、DBEN'、DSF1、DSF2、TRG'およびWE'をそれらの適当な行時間レベルにドライブする。転送コントローラ80はREADYおよびRETRY'入力もサンプリングする。ステート377は、多数回繰り返すことができる。

【0153】外部メモリーインターフェース302は、リフレッシュサイクルに限り、rf6ステート378となる。ステート378中、転送コントローラ80はRAS'出力を低レベルにドライブし、RETRY'入力をサンプリングする。

【0154】rf7ステート379は列タイミング選択CT[1]=1のとき、2および3サイクル/列リフレッシュと共通である。ステート379中、転送コントローラ80はSTATUS[4:0]に関するプロセッサアクティビティコードを出力し、RETRY'入力をサンプリングする。

【0155】外部メモリーインターフェース302は列タイミング選択CT[1:0]=11のとき、3サイクル/列リフレッシュの間に限り、rf8ステート380となる。信号の遷移は生じない。転送コントローラ80はRETRY'入力をサンプリングする。

【0156】rf9ステート381は、すべてのリフレッシュに対する最終ステートである。ステート381中、転送コントローラ80はSTATUS[4:0]に関するプロセッサのアクティビティコードを出力し、RETRY'入力をサンプリングする。

【0157】rhizステート382は、高インピーダンスステートである。外部メモリーインターフェース302は、外部バスホストリクエスト中にステート382となる。バスリリースを介する画像システムバスの復帰までに外部メモリーインターフェース302はステート382を繰り返す。

【0158】列時間ステートは各メモリアクセスのデータサブサイクルまたは行時間を含む。これらステートの間にすべてのデータ転送が行われる。外部メモリーインターフェース302は2つの基本シーケンスの列ステート、すなわちパイプライン化された、またはパイプライン化されていない1サイクル/列アクセスに対するシーケンスと、2および3サイクル/列アクセスに対するシーケンスをとる。ページモードオペレーションの間、マルチ列アクセスが生じ得るので、外部メモリーインターフ

フェース302はシングルページモードアクセスが進行する間、何回もこれらステートシーケンスを繰り返すことがある。

【0159】c1、2ステート382は、列タイミング選択CT[1:0]=10の際の2サイクル/列アクセスのための第1列ステートである。ステート382の間、転送コントローラ80はアドレスバスA[31:0]上の列アドレスおよびSTATUS[4:0]に関するプロセッサアクティビティコードを出力する。転送コントローラ80は、書き込みサイクルであればデータ出力をドライブし、適当なCAS'[7:0]ストロブをアサートする。

【0160】c1、3ステート384は、列タイミング選択CT[1:0]=11の際の3サイクル/列アクセスのための第1列ステートである。ステート384の間、転送コントローラ80はアドレスバスA[31:0]上の列アドレスおよびSTATUS[4:0]に関するプロセッサアクティビティコードを出力する。転送コントローラ80は、書き込みサイクルであればデータ出力をドライブする。

【0161】c2、3ステート385は、列タイミング選択CT[1:0]=11の際の3サイクル/列アクセスのための第2列ステートである。ステート385の間、転送コントローラ80は、適当なCAS'[7:0]ストロブをアサートする。

【0162】c3ステート386は、列タイミング選択CT[1]=1の際の2および3サイクル/列アクセスのための最終列ステートである。ステート386の間、転送コントローラ80は、READY入力をサンプリングし、そのときのサイクルが読み出しサイクルであれば、入力データをラッチする。

【0163】cs1ステート387は、列タイミング選択CT[1]=0の際の1サイクル/列読み出しのための開始列ステートである。ステート387の間、転送コントローラ80は、アドレスバスA[31:0]上の列アドレス、STATUS[4:0]に関するプロセッサのアクティビティコードを出力し、すべての反転CAS'[7:0]ストロブをアサートする。転送コントローラ80は入力データをラッチしない。

【0164】cs2ステート388は第2開始列ステートである。外部メモリアインターフェース302は、列タイミング選択CT[1:0]=00のときに限り、パイプライン化された1サイクル/列読み出しのためにcs1ステート387の後に、常にステート388を挿入する。ステート388中、転送コントローラ80は列アドレスおよび次の列アクセスのプロセッサアクティビティコードを出力する。転送コントローラ80はステート388中に入力データをラッチしない。

【0165】cpステート389は、列タイミング選択CT[1]=0のとき、すべての1サイクル/列アクセ

スに対する基本的列ステートである。外部メモリアインターフェース302は、必要な開始ステートの完了後に各ページモード列アクセスのためにステート389を繰り返す。ステート389中、転送コントローラ80はアドレスバスA[31:0]上の列アドレスおよびSTATUS[4:0]上のプロセッサアクティビティコードを出力し、適当なCAS'[7:0]ストロブをアサートする。転送コントローラ80は有効データの書き込みのためにドライブアウトし、有効データの読み出しのためにラッチインする。

【0166】cdrステート390は、列タイミング選択CT[1]=0のときの1サイクル/列アクセスのための読み出しドレインステートである。外部メモリアインターフェース302は、最終cpステート389の後のステート390またはページ変化またはパイプラインバブルのため、またはインターフェースがアイドル状態となったために、他の列アクセスがペンディング中でない場合、cs1ステート387を有する。ステート390中、転送コントローラ80は反転CAS'[7:0]ストロブをアサートしないが、先のcpステート389またはcs1ステート387によりアドレス指定される入力データをラッチする。

【0167】外部メモリアインターフェース302は、列タイミング選択CT[1:0]=00のとき、パイプライン化された1サイクル/列書き込みドレインサイクルのためのcdwステート391となる。外部メモリアインターフェース302は新しいページの開始時にr1ステート370に復帰する直前に、ステート391となる。ステート391中、転送コントローラ80はすべてのCAS'[7:0]ストロブをスタートし、最終書き込みされたデータでパイプライン化されたメモリアレイを更新する。

【0168】ciステート392は、アイドル列ステートである。必要なドレインステートの後にそれ以上の列アクセスがペンディング中でなければ、外部メモリアインターフェース302はステート392となる。これはパイプラインバブルによって生じ得るか、またはそれ以上のサイクルがリクエストされておらず、ページ変化が生じないためインターフェースがページモードに留まっているからである。

【0169】図10に示したステート遷移インディケータは、どの条件によって別のステートへの遷移が生じるかを決定する。この場合、所定のステートへの遷移が生じるには、多数の条件を満たさなければならない。これらステート遷移条件および事象のリストは次の通りである。

【0170】CT=XXr3ステート372の間にラッチされた表示列タイミング選択CT[1:0]の値に対してステート変化が生じる。

【0171】

r、読み出しー読み出しサイクルが生じる。

w、書き込みー書き込みサイクルが生じる

r tー読み出し転送サイクルが生じる。

w tー書き込み転送サイクルが生じる。

フォールトー r 3 ステート中に低レベルにサンプリングされた FAULT' 入力、メモリアクセスがフォールトであることを表示する。

再試行ー r 3 ステート 372 に低レベルにサンプリングされた RETRY' 入力、行時間再試行をリクエストする。

待機ー r 3 ステート 372、r 6 ステート 377 または c 3 ステート 386 中に低レベルにサンプリングされた READY 入力は、そのときのステートの繰り返しをリクエストする。

スピンーこれは転送コントローラ 80 のパイプラインがロードできるように、別の r 3 スピン 377 を加えるため内部で発生されたリクエストである。外部メモリインターフェース 302 は、先のページアクセスからバスサイズが変わると、各 2 サイクル/列書き込み中に 1 回、各 1 サイクル/列各中に 2 回、1 サイクル/列アクセス中に 1 回、ステート 372 となる。

新しい列ーこのときのメモリサイクルは同じメモリページの次の別アクセスである。

ドレインーこのときのメモリサイクルは 1 サイクル/列読み出しサイクルである。このときの読み出しステートの後には、ペンディングとなっている列アクセスはない。

アイドルーそのときのステートの後にペンディングである列アクセスは無く、すべてのドレインステートは完了している。新ページー次のメモリアクセスはページの変更および新しい行アクセスを必要とする。

【0172】転送コントローラ 80 は内部メモリインターフェース 301 および外部メモリインターフェース 302 の双方のうちにパイプラインを含む。そのときのアクセスがまだ完了していないとき、これらパイプラインは転送コントローラ 80 が必要とするメモリアクセスを待つことができる。例えばパケットリクエストがオンチップメモリからオフチップメモリに転送中である場合、宛て先サイクルはアクセス当たり 2 サイクルを必要とすることがある。必要とされるデータがパケット転送 FIFO バッファ 311 にあると仮定すると、アクセス当たり 1 サイクルでパケット転送 FIFO バッファ 311 からデータを抽出できるので、第 1 サイクルが完了する前に、パイプラインに別の宛て先サイクルを挿入できる。

【0173】これらパイプラインは通常ユーザーに対して完全にトランスペアレントである。作動に対するこの効果は、パイプラインがドレイン状態となっているときに見えるだけである。外部メモリインターフェース 302 の外部メモリパイプラインに 1 サイクルをロードするには、パイプラインに既に含まれている他のサイクルと

同じメモリページ内にこのサイクルが入っていなければならない。パイプラインに一旦 1 つのサイクルを挿入すると、これを除くことはできず、このサイクルを発生しなければならない。従って新しいメモリページへのアクセスがリクエストされると、まず最初にパイプライン内のサイクルを完了しなければならない。転送コントローラ 80 が外部メモリへのパケット転送を行い、フレームコントローラがメモリリクエストを発生する場合、フレームコントローラサイクルがより高い優先度を有していたとしても、フレームコントローラ 90 がリクエストしたサイクルを発生する前に、現にパイプライン内にある列サイクルを完了しなければならない。キャッシュおよび外部メモリアクセス、ホストアクセスおよび緊急リフレッシュ前、パケット転送一時中断中、および列時間再試行後にも、パイプラインのドレインを行うことができる。緊急リクエストがペンディングのとき、パイプラインのドレインには最高の優先度が与えられる。

【0174】外部メモリアクセス中、作動サイクルが存在していてもよく、この間にアクティブな列アクセスは生じない。これらの期間は転送コントローラ 80 の内部パイプライン内のバブルの結果である。転送コントローラ 80 がこのサイクル中に実行される動作がないときにバブルが生じる。これは転送コントローラ 80 が書き込みサイクルのための次のアクセスを実行するのに十分なデータを有していないこと、パケット転送 FIFO バッファ 311 がフル状態であり、読み出しサイクルでそれ以上データをロードできないこと、また、端にアクティビティのリクエストはないことを理由とする、クロスバーの競合の結果として生じ得る。これら条件のいずれも生じない場合、バブルも発生しない。

【0175】バブルの一例は、転送コントローラ 80 が内部ソースおよび外部宛て先のパケット転送のサービスをしているときである。転送コントローラ 80 があるときにソースデータの 1 バイトにしかアクセスできず、そのときに書き込み宛て先データの 8 バイトしか書き込みできないように、パケット転送がセットされている場合、転送コントローラ 80 は 8 回の内部サイクルごとに 1 回、宛て先アクセスを行うことができるだけである。この結果、転送コントローラ 80 は次の列アクセスを実行するのに十分なデータの 8 バイトを持つのを待ちながら、外部メモリ宛て先サイクル内にバブルを挿入する。挿入されるバブルサイクルの回数は、外部メモリサイクルのタイミングに依存する。

【0176】広範なメモリのタイプおよび組み合わせ、更に周辺機器をサポートするため、マルチプロセッサ集積回路 100 は外部メモリサイクルの拡張、再試行および終了のための種々の方法を提供する。これらの各々について次の章でより詳細に説明する。

【0177】転送コントローラ 80 がパイプライン内でペンディング中の列アクセスのすべてを完了すると、メ

モリアクセスの終了の準備ができる。しかしながら新しい行アクセスが必要となるまで終了は行われない。次のメモリアクセスまでc iステート392において外部メモリ信号はアクティブ状態のままである。これによりDRAMデバイスはページモードステートの状態に維持できる。次のアドレスが先のメモリアクセスと同じメモリページ内に入り、同一方向である場合、行アクセスサイクルは不要である。次のメモリアクセスが行アクセスを必要とする場合、そのときのページモードアクセスは終了し、新しい行アクセスが開始する。それ以外の外部バスアクティビティがほとんどない場合、新しい行アクセスの発生を必要とするトリックリフレッシュサイクルの発生により、最終列アクセスの終了後にすぐにほとんどのメモリサイクルが終了される。

【0178】マルチプロセッサ集積回路100はメモリサイクル時間を延長するように、待機ステートの挿入をサポートする。これはREADY入力を用いることによって行われる。このREADY入力は、CLKOUTの効果エッジで適当な時間にサンプリングされたものであり、READYがハイレベルでサンプリングされた場合、サイクルは通常の態様で続行する。READYが低レベルでサンプリングされた場合、現在のマシンステートが繰り返され、次のCLKOUTで再びREADYがサンプリングされる。このメモリサイクルは転送コントローラ80がハイレベルのREADYをサンプリングするまで、そのときのステートを繰り返すように続く。

【0179】転送コントローラ80は多数の待機ステートによりホールドアップされるメモリアクセスを終了させるようなタイムアウトまたはアボート機構を有していない。サービスを待つことができる高い優先度の外部メモリアクセスのロックアウトを防止するため、適当な時間に完了できないメモリアクセスはフォールトとするか、または再試行すべきである。

【0180】任意のタイプのメモリサイクルに対する行時間の間に、待機ステートを加えることができる。列アクセス当たり2サイクルおよび3サイクルの双方の間で、列時間において待機ステートを挿入することもできる。待機ステートは反転RASの降下の前後の行時間の間に2つの位置で待機ステートを挿入できる。適当なコントローラ80は、まずr3ステート372またはrf3ステート373またはtr3ステート374の開始時に、READY信号をサンプリングできる。これによりこの時間で行アドレスおよび/またはサイクルタイプをデコードし、反転RASの降下前にアドレス指定されたデバイスが、更なるアクセス時間を必要とするかを判断できる。転送コントローラ80がハイレベルのREADYをサンプリングするまで、r3ステート372が繰り返される。READYが低レベルでサンプリングされるとき、r3ステート372が繰り返されるが、反転FAULTおよび反転RETRY入力は再サンプリングされ

ない。待機ステートが挿入される前の第1r3ステート372サイクル中にアクセスをフォールトするか、または再試行しなければならない。r6ステート377中でREADY入力を再びサンプリングする。これはRAS'の降下後に起きるので、必要とし得るデバイスに対し、追加RAS'アクセスタイムを発生できる。READYが低レベルでサンプリングされる場合、このREADYが再びハイレベルでサンプリングされるまでr6ステート377が繰り返される。

【0181】転送コントローラ80は2および3サイクル/列書き込みおよびバスサイズを変更する1サイクル/列アクセスの間に自動的に付加的r6ステート377を挿入する。転送コントローラ80はこれらr6ステート377の各々の間で通常の態様でREADYをサンプリングする。従って、低レベルでサンプリングされるREADYから生じる余分なr6ステート377は、転送コントローラ80によって既に挿入されたステートに加えられる。

【0182】列アクセス当たり1サイクルの間では、列時間待機ステートはサポートされていないので、これらサイクルの間でr6ステート377を越えてREADYはサンプリングされない。転送コントローラ80は、列アクセス当たり2および3サイクルに対しc3ステート386の間でREADYピンをサンプリングする。これは列アドレスおよび反転CASの降下後に生じるので、必要とするデバイスに対し延長されたCASアクセス時間を与えることができる。READYがハイレベルでサンプリングされる場合、c3ステート386は列アクセスを完了させる。そうでない場合、c3ステート386が繰り返され、ハイレベルでサンプリングされるまで転送コントローラ80はその後のCLKOUT降下エッジごとにREADYをサンプリングする。

【0183】マルチプロセッサ集積回路100は2つのタイプの再試行機構、すなわち行時間再試行および列時間再試行をサポートしている。行われる再試行のタイプは、マルチプロセッサ集積回路100の反転RETRY入力がアクティブ低レベルとなるメモリアクセスのどの点にあるかによって決まる。

【0184】行時間再試行は、現在のアクセスが完了できず、再試行が必要なことを外部ロジックがマルチプロセッサ集積回路100に伝えることができる方法を提供する。この行時間再試行は、r3ステート372の開始点でまずサンプリングされる際、RETRY'入力を低レベルにドライブすることによって発生される。発生RETRYが低レベルでサンプリングされる場合、転送コントローラ80はr3ステート372の終了点で現在のアクセスを終了する。次により高い優先度のリクエストがペンディング中でなければ、再試行されるアクセスは新しいr1ステート370と共にすぐに再スタートする。この場合、再試行されるサイクルは、より高い優先

度のリクエストのサービスが完了するまで再スタートされない。

【0185】パケット転送サイクル中に行時間再試行が行われ、同等以上の優先度のパケット転送リクエストがペンディング中であれば、行われている再試行が一時中断され、PTMINによって定義されるインターバルが終了したとみなされる。パケット転送が優先度/ラウンドロビンチェーン内の順番に再び達し、パケット転送が再開されるまで、転送コントローラ80は、この一時中断されたパケット転送を再試行することはない。

【0186】列時間再試行は従来の意味では実際には再試行ではない。むしろこれらは行アクセスリクエストと考えることができる。列再試行が生じると、転送コントローラ80はそのパイプライン内でペンディング中のすべての列アクセスを完了し、行アクセスと共に次のアクセスを開始する。従って再試行がリクエストされた後に多数の列アクセスが行われることがある。行アクセスが再スタートされた後にはこれら行アクセスは繰り返されることはない。r3ステート372の後で反転RETRY入力が低レベルでサンプリングされる時間に列時間の再試行が行われる。転送コントローラ80は各CLKOUT降下エッジでr3ステート372の後の各ステートの終了時点で、RETRY'をサンプリングするので、反転RETRYは各サンプリング期間中で有効な高レベルまたは低レベルとなっていなければならない。転送コントローラ80が低レベルの反転RETRYをサンプリングする場合、転送コントローラ80のパイプライン内に現在あるすべての列アクセスが完了するとすぐに、そのときのページモードが終了する。再試行が一旦リクエストされると、反転RETRY入力に入力された値はそれ以上の効果がなくなる。しかしながら転送コントローラ80は残りの列アクセスの各々の間で反転RETRYをサンプリングし続けるので、RETRY'は各サンプリング期間中に高または低レベルに維持されていなければならない。通常のオペレーションは、システムがRETRY'を低レベルにドライブし、現在の行アクセスの終了時までこれを低レベルに維持させるためである。

【0187】外部メモリインターフェース302のパイプラインにロードする方法により、読み出しサイクル中にr3ステート375またはr5ステート376でRETRY'をアサートすることは全く効果がない。RETRY'入力はまだサンプリングされており、有効な高または低レベルにななければならない。このため、列時間再試行が望まれる場合、行アクセスの終了時までRETRY'を低レベルにアサートしなければならない。フレームコントローラ90のVRAMアクセスサイクルおよびリフレッシュサイクルは、列時間によって影響されないが、その理由は、性質上、新しい行アクセスは単一行アクセスの後に続くからである。

【0188】システムエラーのため、システムがメモリ

アクセスを完了できない場合、システムはメモリサイクルをフォールトすることによりマルチプロセッサ集積回路100に通知できる。これによりメモリサイクルを再試行する前にマスタプロセッサ60はエラーを訂正できる。メモリフォールトは行時間で発生できるだけであり、r3ステート372の開始時にFAULT'入力を低レベルにドライブすることにより開始される。転送コントローラ80はメモリサイクルの他の部分の間で反転フォールトをサンプリングすることはない。フォールト機構はフォールトを生じさせたアクセスのタイプに多少依存して変わる。これについては後により詳細に説明する。異なるタイプのサイクルリクエストに対するメモリフォールトは次のようにサポートされている。

【0189】フレームコントローラ90のサイクル。フレームコントローラ90のサイクルの間にフォールトはサポートされておらず、反転FAULTピンは無視される。リフレッシュサイクル。リフレッシュサイクル中にフォールトはサポートされず、反転FAULTピンは無視される。

【0190】デジタル画像/グラフィックプロセッサキャッシュおよびダイレクトメモリアccessリクエスト。これらサイクルの間にフォールトはサポートされる。リクエスト中のプロセッサはフォールトがクリアされるまで完了されるリクエストは見ない。他のデジタル画像/グラフィックプロセッサ71、72、73、74からのリクエストはサービスされ続ける。

【0191】マスタプロセッサのインストラクションまたはデータキャッシュリクエスト。これらサイクル中にサポートされるフォールトはサポートされる。フォールトされたキャッシュリクエストは即座にキャンセルされ、マスタプロセッサ60はインタラプトされる。その他のキャッシュはペンディング中のリクエストをサービスすることができる。

【0192】マスタプロセッサ60のダイレクトメモリアccessリクエスト。これらサイクル中にはフォールトがサポートされる。外部アクセスリクエストが即座にキャンセルされ、マスタプロセッサ60がインタラプトされる。

【0193】パケット転送。パケット転送なかにはフォールトがサポートされる。このパケット転送は一時中断され、そのステートはリクエスト中のプロセッサのパラメータメモリにセーブされる。他のプロセッサからのパケット転送リクエストはまだサービス可能である。セーブされたパラメータ内には転送コントローラ80の内部パケット転送ステートが含まれる。パケット転送が外部メモリ間で外部メモリなされる場合、外部メモリ間の転送バッファのステートもセーブされる。バッファ自体は変わらない。セーブされたパケット転送オプションフィールド内のパケット転送ステートビットは、フォールトがソースへの転送または宛て先への転送で生じたかどうか

かを示すのにセットされる。

【0194】一旦、パラメータがセーブされると、転送コントローラ80はマスタプロセッサ60のうちのFLTSTSレジスタ内の適当なビットをセットし、どのプロセッサのパケット転送がフォールトしたかを表示する。下記の表13にはFLTSTSレジスタのコーディングが示されている。FLTSTSレジスタ内のプロセッサビットもマスタプロセッサ60に対し、パケット転送フォールトインタラプトを発生する。マスタプロセッサ60はどのプロセッサがフォールトされたパケット転送をリクエストしたかを探すため、FLTSTSレジスタを読み出すことができる。一旦プロセッサが識別されれば、マスタプロセッサ60は一時中断されているパケット転送パラメータを検査し、フォールトを生じさせたメモリアクセスを決定できる。

【0195】パケット転送中にフォールトが生じたことは、デジタル画像／グラフィックプロセッサ71、72、73または74は知らない。このプロセッサは、パケット転送がまだ完了していないことを知っているだけである。フォールトを訂正したり、またはリクエスト中のデジタル画像／グラフィックプロセッサ71、72、73または74に、そのパケット転送リクエストのキャンセルを求めるのは、マスタプロセッサ60の役割である。

【0196】マスタプロセッサ60がフォールトを訂正できる場合、FLTSTSレジスタ内のビットをクリアすることができ、パケット転送リクエストは自動的に再送信される。フォールトしたパケット転送は、ラウンドロビン優先法における順番を受けると、その内部ステートがパラメータメモリ内のセーブされたパラメータからレストアされ、転送コントローラ80はフォールトされたアクセスでパケット転送を続ける。

【0197】デジタル画像／グラフィックプロセッサがリクエストしたキャッシュサービスまたはダイレクト外部メモリアクセスリクエスト中にフォールトが生じた場合、リクエスト中のプロセッサのパラメータメモリのキャッシュフォールトアドレス位置に、フォールトの生じたアドレスがセーブされる。マスタプロセッサ60のFLTSTSレジスタにおける適当なビットがセットされ、インタラプトがマスタプロセッサ60に送られる。マスタプロセッサ60がパラメータメモリを検査し、フォールトしたアドレスを決定できる。マスタプロセッサ60がフォールトを訂正できる場合、このプロセッサはFLTSTSレジスタ内のビットをクリアし、リクエストは再スケジュールされる。フォールトが訂正できず、マスタプロセッサ60がリクエスト中のデジタル画像／グラフィックプロセッサにキャッシュミスまたはダイレクト外部アクセスリクエストをアボートすることを求める場合、デジタル画像／グラフィックプロセッサにリセットリクエストを送らなければならない。これによりそ

のタスクがアボートされる。

【0198】デジタル画像／グラフィックプロセッサ71、72、73または74は、キャッシュまたはダイレクト外部アクセスリクエスト中にフォールトが生じたことを知らず、そのリクエストが完了していないことを知っているにすぎない。フォールトを訂正したり、リクエスト中のデジタル画像／グラフィックプロセッサをリセットするのは、マスタプロセッサ60の役割である。

【0199】マスタプロセッサ60がキャッシュフィルまたはダイレクト外部アクセスサイクルのフォールトをリクエストした場合、このリクエストは即座にキャンセルされ、マスタプロセッサ60にメモリフォールトインタラプトが送られる。このメモリフォールトインタラプトはフォールトがデータキャッシュフォールトまたはインストラクションキャッシュフォールトであったことを表示する。データキャッシュフォールトが生じた場合、そのアドレスがセーブされ、データがセーブされる。

【0200】オンチップアドレスへの所定のアクセスは、反転FAULT入力と独立したフォールトを生じさせ得る。不法なオンチップアクセスが行われると、これが生じる。試みられたサイクルに対する通常のフォールト機構が適応される。オンチップフォールトとは次のとおりである。

【0201】データメモリまたはデジタル画像／グラフィックプロセッサパラメータメモリでない16進数02000000以下のアドレスとの間でのデジタル画像／グラフィックプロセッサのパケット転送。

【0202】データメモリまたはデジタル画像／グラフィックプロセッサパラメータメモリでない16進数02000000以下のアドレスとの間でのデジタル画像／グラフィックプロセッサのキャッシュサービス、またはマスタプロセッサ60のキャッシュサービス、またはダイレクト外部アクセス。データメモリまたはデジタル画像／グラフィックプロセッサパラメータメモリでない16進数02000000以下のマスタプロセッサ60のパケット転送。

【0203】マルチプロセッサ集積回路100によって発生されるメモリタイミングと異なるメモリタイミングの発生を望むユーザーを補助するために、UTIME'が発生される。このUTIME'が行時間において低レベルでサンプリングされると、現在のページの残りに対して、反転RASおよびCAS'〔7:0〕出力のタイミングが変えられる。列アクセスが開始したことを表示するように、RAS'信号が変えられる。従ってRAS'の降下によってユーザーがタイミングを決めたCAS'信号をトリガできる。実際の列アクセスが開始されたときに、マシンスタートのみににおいてRAS'をアクティブ低レベルにアサートできるにすぎないので、転送コントローラ80のパイプライン内のバブルを理解することができる。行アドレスと同時に列アドレスストロー

ブCAS' [7:0] が出力される。これにより、これらストロープはどのバイトがアクセスされたかを表示するのに使用でき、外部CAS' の発生が容易となる。RAS' のタイミングは変わっているので、必要な場合には外部ロジックが自らのRAS' タイミングを発生しなければならない。外部から発生されたRAS' は、RL' の降下エッジによってトリガできる。サイクルの開始時点で出力されたステータスコードは、そのときのサイクルの間にてTRG'、WE' 等のエモリタイミングを発生するのに必要なすべの情報を提供することに留意されたい。

【0204】転送コントローラ80の内部メモリアンターフェース301は、クロスバー50を介し、オンチップメモリにアクセスする。64ビットの内部データバスは、1サイクルにつき0～8バイトを転送できる。外部メモリアンターフェース302は、オフチップメモリおよび周辺機器のすべてにアクセスする。外部メモリアンターフェース302は、種々のメモリおよび周辺機器のタイプにインターフェースするのに必要なサイクルおよび制御信号を発生する。64ビットの外部データバスは1サイクルにつき0～8バイトを転送し、8、16、32および64ビットのデータ幅を有する画像システムバスに接続されたデバイスに対し、ダイナミックバスサイズのサポートをする。

【0205】外部メモリアンターフェース302は、転送コントローラ80が外部デバイスとバスを共用できるようにする簡単なハンドシェイク機構を含む。このハンドシェイク機構は4つの同期信号を活用する。すなわちホストリクエスト入力HREQ'、ホストアクノリッジ出力HACK' および内部リクエスト出力RREQ [1:0] である。

【0206】外部デバイスが画像システムバスの制御を望む際、ホストリクエスト入力HREQ' は、マルチプロセッサ集積回路100に信号を送る。外部デバイスはバスのオーナーシップを持ちたい場合、ホストリクエスト入力HREQ' を低レベルにドライブする。これは転*

*送コントローラ80が受けることもできる最高優先度のリクエストであり、転送コントローラ80は、最も初期に起こり得る時間にバスのドライブを停止させる。そのときのオペレーションが終了し、転送コントローラ80のパイプラインが空になると、転送コントローラ80のバスオーナーシップが終了する。外部デバイスはバスを望む限り、HREQ' を低レベルアクティブにドライブし続けなければならない。HREQ' がハイレベルで非アクティブとなると、転送コントローラ80はバスを所有し、これをドライブする。HREQ' 入力は内部的にマルチプロセッサ集積回路100の内部ブロックに同期している。

【0207】ホストアクノリッジ出力HACK' は、転送コントローラ80が画像システムバスの制御信号を発生する準備が完了した信号を発生する。転送コントローラ80はアクティブなHREQ' の後に、HACK' を低レベルにドライブし、その信号を高インピーダンスにドライブし、バスを放棄していることを表示する。反転HACKが低レベルアクティブであるとき、CLKOUTを除くすべての外部メモリアンターフェース302のバス出力が、高インピーダンスとされる。内部リクエスト出力REQ [1:0]、HACK' およびフレームコントローラ90の出力は、ドライブされ続ける。外部デバイスは必要に応じて画像システムバスをドライブできる。転送コントローラは、HREQ' が非アクティブとなったことが検出された後に、非同期的にHACK' をハイレベルで非アクティブにドライブし、バスのドライブを再開する。

【0208】内部リクエスト出力REQ [1:0] は、転送コントローラ80によって受信される最高優先度の内部リクエストの2つのビットコード化を形成する。表3にリクエストコードおよびそれらの関連するサイクルを示す。

【0209】

【表10】

REQ [1:0]	関連内部リクエスト
0 0	低優先度パケット転送 トリクルリフレッシュ アイドル
0 1	高優先度パケット転送
1 0	インストラクションキャッシュサービス データキャッシュサービス
1 1	MP緊急パケット転送 フレームコントローラアクセス 緊急リフレッシュ

表10

【0210】外部ロジックはいつバスを放棄して転送コントローラ80に戻すかを決定するため、内部リクエスト出力REQ [1:0] をモニタできる。転送コントロ

ーラ80によってホストリクエストには最高の優先度が与えられているので、システムデザイナーは、ホストがバスの所有権を放棄して転送コントローラ80に戻すの

はどのREQ[1:0]の値にするかを決定することにより、外部デバイスが作動するレベルを決定できる。

【0211】転送コントローラ80は外部メモリシステムが必要とするDRAMリフレッシュサイクルを自動的に発生するためのプログラマブルリフレッシュコントローラを含む。リフレッシュレジスタ313に、このリフレッシュのためのパラメータが記憶される。キャッシュ制御ロジックは、マスタプロセッサ60およびデジタル画像/グラフィックプロセッサ71、72、73および74がリクエストするようなキャッシュフィルおよび書き込みバックを実行するのに必要なアドレスを発生する。フレームコントローラ90のリクエストはVRAM制御論理によって処理される。

【0212】転送コントローラ80はパケット転送を処理するための2つの独立したコントローラを有する。ソースレジスタ321とソースコントローラ322から成るソースマシンは、ソースメモリからデータをフェッチするのに必要なアドレスを発生する。転送コントローラ80にパケット転送リクエストを送る際、このリクエストはソースデータをどのようにアクセスすべきかを指定する多数のパラメータを含む。これらパラメータはソースレジスタ322にロードされ、ソースアドレスを発生するようソース制御ロジック323によって使用される。同様なパラメータの組が宛て先レジスタ341にロードされ、宛て先制御ロジック342によって使用され、宛て先メモリ領域にパケットデータを書き込むのに必要なアドレスを発生する。宛て先レジスタ341と宛て先制御ロジック342の組み合わせにより宛て先マシン340を形成している。ソースマシン320および宛て先マシン340は各々オンチップメモリ10および20とオフチップメモリとの双方のアドレス指定が可能である。

【0213】図11は、ソースマシン320の構造を示す。ソースマシン320の内部構造と宛て先マシン340の構造は同一である。転送コントローラ80の他に対する接続のいくつかが異なっているにすぎない。ソースマシンは次のようなレジスタを含む。CACHEADRレジスタ401は転送方向に応じて内部アドレスまたは外部アドレスのいずれかを含む。このアドレスは、キャッシュサブブロックのアドレスと共に、転送コントローラ80の内部Sポートからロードされる。オンチップアドレスの上部ビットを発生するのに、マルチプレクサを使用できる。

【0214】ABPITCHレジスタ402は、次のラインのための開始アドレスを発生するため、開始アドレスに加算される（または減算される）ピッチを含む。GTPPIPEレジスタ403は実際には3つの部分に分割されている。第1部分は、ASTARTが宛て先となっている32ビットアドレスである。第2部分はBCOUNTが宛て先となっている16ビットのラインカウン

であり、第3部分はACOUNTが宛て先となっている16ビットのバイトカウントである。このレジスタは、関連するレジスタにフェッチしたガイドテーブルを即座に記憶できないとき、ガイドテーブルオペレーション中の一時レジスタとして働く。

【0215】ASTARTレジスタ404は、現在のラインの開始アドレスの記録を記憶し、ASTARTレジスタ404は、ガイドされた転送を実行する際にガイドテーブルアドレス/オフセット入力に対して倍になる。BSTARTレジスタ405は、現在のパッチの開始アドレスの記録を記憶する。BSTARTレジスタ405は、また、オフセットガイドされた転送に対しベースアドレスを含み、デルタガイドされた転送に対しては最終パッチ開始アドレスを含む。

【0216】CRNTADRレジスタ406はパケットサービスの現在のアドレスを記憶する。このアドレスは次のアクセスを実行する場所のアドレスである。次のアドレスを発生するよう転送されるバイト数がこの値に加算されたり、またはこれより減算される。ラインまたはピッチが完了すると、CRNTADRレジスタ406には次のラインまたはパッチの開始アドレスがロードされる。

【0217】BCPITCHレジスタ407は、次のパッチのための開始アドレスを発生するよう、現在のパッチの開始アドレスに加算もしくはこれより減算されるピッチを記憶している。ガイドされた転送を行うとき、このBCPITCHレジスタ407は、ガイドテーブルポインタとして使用される。ACOUNTレジスタ408はライン内のバイト数を記憶している。このバイト数は定数であるか、または可変パッチガイドされた転送のためのガイドテーブル入力からロードされる。ACURRENTレジスタ409はライン内に残っているそのときのバイト数を記憶する。

【0218】BCOUNTはパッチ内のライン数を記憶する。この数は定数であるかまたは可変パッチのガイドされた転送のためのガイドテーブル入力からロードされる。BCURRENTレジスタ411は、パッチ内に残っている1未満のラインの現在の数を記憶する。CCURRENTレジスタ412は、パケット内に残っている1未満のパッチの現在の数を記憶する。ガイドされた転送を行うとき、これはその代わりにガイドテーブル内に残っている入力の数を表示する。

【0219】CBYTESレジスタ413はダイレクトメモリアクセスまたはキャッシュサービスにより転送すべき残っているバイトの現在の数を記憶する。BUFPTRレジスタ414は、ソースマシン320と宛て先マシン340によって共用されるレジスタである。BUFPTRレジスタ414は、外部メモリ間でのパケット転送中にバッファとして使用されるパラメータメモリの部分をポイントする。マルチプレクサはオンチップアドレ

スの上部ビットを発生するのに使用できる。

【0220】FCOUNTレジスタ431（図12に示す）は、パイプライン内に現在あるすべてのインストラクションが完了した後、パケット転送用FIFOバッファ311内にどれだけ多くのバイトが残るかを示す5ビットの値（16～0のみ）を記憶する。

【0221】BUFCOUNTレジスタ441（図13に示す）は、外部メモリ間バッファ内にどれだけ多くのスเปアバイトが残るかを示す8ビットの値（128～0のみ）を記憶する。ソースマシン320の中心は、マシンが転送したいバイト数を決定するための計算用ハードウェアと方法である。これはバイト計算回路415の数*

3つの最小位 のアドレス ビット	64ビットバス		32ビットバス		16ビットバス		8ビットバス	
	前方	後方	前方	後方	前方	後方	前方	後方
000	8	1	4	1	2	1	1	1
001	7	2	3	2	1	2	1	1
010	6	3	2	3	2	1	1	1
011	5	4	1	4	1	2	1	1
100	4	5	4	1	2	1	1	1
101	3	6	3	2	1	2	1	1
110	2	7	2	3	2	1	1	1
111	1	8	1	4	1	2	1	1

表11

【0223】バスが64ビット幅であり、アドレス指定が前方であれば、3つの最小位のアドレスビットの2の補数をとることにより、上記の値を推定する。4番目のアドレスビットは桁上げ用である。次元が後方にアドレス指定される場合、3つの最小位のアドレスビットの値に1を加えるだけでバイト数を計算する。バスが32ビット幅で、アドレス指定が前方であれば、2つの最小位のアドレスビットの2の補数をとることにより、上記の値を推定する。桁上げにより第3ビットとなる。次元が後方にアドレス指定される場合、2つの最小位アドレスビットの値に1を加えるだけで、バイト数を計算できる。これらの場合、計算に通常含まれる次の最高位ビットの影響は無視しなければならない。バスが16ビット幅で前方にアドレス指定される場合、最小位アドレスビットだけの2の補数をとることにより、上記値が推定される。桁上げは第2ビットとなる。次元が後方にアドレス指定される場合、最小位ビットの値に1を加えるだけでバイト数が計算される。これらの場合、通常計算に含まれる次の2つの最高位ビットの影響は無視しなければならない。

【0224】上記の説明は、マシンがそれ自体を現在のバスサイズに一致させるため、転送を望むバイトの数について述べたものである。このオペレーションが行われた後に、連続する転送は自動的に現在のバスサイズによって可能なバイトの最大数となる。このバスサイズを変更する場合、これも自動的に処理される。次にアドレスレジスタはバイト数だけインクリメントまたはデクリメントされ、次のデータの開始点をポイントする。現在の次元カウンタの残りのバイトの新しい数を示すように、

*で行われる。これは次元内に残っているバイト数、アドレスの一致、バスサイズおよび次元が前方にアドレス指定されているか、または後方にアドレス指定されているかどうかに応じて決まる。エンディアンはこの計算には重要ではない。次元内に残っているバイト数が8以上であると一時的に仮定すると、表11は異なっているアラインメントのためにマシンが転送を望んでいるバイト数、バスサイズおよびアドレス指定の方向（前方または後方）を示している。

【0222】

【表11】

同じ数だけデクリメントされる。マシンが転送を望むバイト数が、次元内に残っているバイト数よりも大きくなるまでこのプロセスが続き、残りのバイト数よりも転送を望むバイト数が大きくなった時点まで、実際に転送されるバイト数を低い値にしなければならず、こうして次元の転送を完了する。このような条件は、転送を望むバイト数から残っているバイト数を減算することにより検出され、借りが起こらない場合は次元の最終転送に達し、転送すべきバイト数のためにバイトカウンタを用いる。借り条件に基づくこれら2つの値の選択のためにマルチプレクサが使用される。

【0225】転送が次元内の最終であり、パケットリクエストが実行中であれば、通常の第1次元計算を行う代わりに、次の次元パラメータを計算する。BCURRENTレジスタ411内の値が0であるかどうかにより、BCURRENTレジスタ411またはCCURRENTレジスタ412のいずれかをデクリメントする。いずれの場合にせよ、xCOUNTレジスタから小さい次元のxCURRENTカウンタをロードし、次の第1次元を開始する。

【0226】BCURRENTレジスタ411の値が0となるかどうかに応じて、ASTARTレジスタ404にABPITCHレジスタ402に記憶されていた値を加算（または減算）するか、またはBSTARTレジスタ404にBCPITCHレジスタ407に記憶されていた値を加算（または減算）することにより、次の次元のためのアドレスも計算される。計算された値は、CRNTADRレジスタ406およびASTARTレジスタ404に書き込まれBCURRENTレジスタ411内

の値が0であれば、BSTARTレジスタ405にも書き込まれる。

【0227】第1次元の最終転送が実行され、BCURRENTレジスタ411およびCCURRENTレジスタ412内の値がいずれも0であれば、ソースマシン320はパケットサービスを完了する。ソースマシン320および宛て先マシン340の双方がこの状態に達すると、全パケットサービスが完了する。パケット転送オプションフィールドのソースおよび宛て先更新モードに応じ、ステージで追加次元計算を行うことができ、CRNTADRレジスタ406に記憶されていたデータが下のパケットリクエスト位置にセーブされる。これにより残されている点でのパケット転送の後における再送信が可能となる。

【0228】これらアドレス計算は2つの32ビット加算器420および423で行われる。マルチプレクサ421および422は、加算器420への入力を選択し、同様にマルチプレクサ424および425は加算器423への入力を選択する。加算器420は、CACHEADRレジスタ401、ABPITCHレジスタ402、ASTARTレジスタ404、BSTARTレジスタ405、CRNTADRレジスタ406およびBCPITCHレジスタ407のサービスを行う。加算器423は、ACOUNTレジスタ408、ACURRENTレジスタ409、BCOUNTレジスタ410、BCURRENTレジスタ411、CCURRENTレジスタ412およびCBYTESレジスタ413に対するサービスを行う。

【0229】上記説明は、パケット転送について簡単に触れたものであるが、同様な方法がキャッシュサービスおよびバッファ操作に対しても適用されることに留意されたい。キャッシュサービスの場合、アドレスは常に一致した境界上で開始し、アドレスがキャッシュサブブロックの終了点に達するとサービスが終了するので、バイトカウンタはない。このことは、ソースマシン320がデジタル画像/グラフィックプロセッサのインストラクションキャッシュまたはマスタプロセッサ60のデータまたはインストラクションキャッシュにサービスするかどうかに応じて変わる。

【0230】上記説明は、転送するバイト数の計算を常に実行できることを示唆している。パケット転送用FIFOバッファ311は、マシンがその数のバイトを転送するには、過度に満杯または空状態となり得るので、このことは当てはまらない。したがってパイプライン内にオペレーションをロードする前にさらに計算を行う必要がある。

【0231】ソースマシン320がパケット転送用FIFOバッファ311内に残されている余裕よりも多いバイトを転送したい場合、待機する必要がある。宛て先マシン340がパケット転送用FIFOバッファ311内

にあるバイト数よりも多いバイトを転送したい場合も待機する必要がある。宛て先マシン340が読み出しをするのに、パケット転送用FIFOバッファ311内に十分なバイトをソースマシン320がロードしようとする場合、双方の転送を続けることができる。

【0232】これら計算は図12に示されたロジックで実行される。FCOUNTレジスタ431は既にパイプライン内にあるオペレーションが完了した場合、パケット転送用FIFOバッファ311内に入るデータのバイト数を記録する。加算器434はソースマシン320がパケット転送用FIFOバッファ311に転送したいバイト数にこの値を加える。合計値が16を越える場合、ソースマシン320は続けることはできないが、宛て先マシン320がいくらかのデータを空にすることを待たなければならない。加算器434は、17個以上の出力を発生し、この出力はソースマシン320に供給され、このマシンをストールさせる。現在の値または加算器434の17以上の出力により選択された加算の結果のいずれかであるパケット転送用FIFOバッファ311内の新しいバイト数がマルチプレクサ435を通して減算器436へ送られる。減算器436は宛て先マシン340が転送したいバイト数からパケット転送用FIFOバッファ311内の新しいバイト数を減算する。減算器436が借り信号を発生しなければ、宛て先マシン340は続行できる。減算器426が借り信号を発生すると、この借り信号はマルチプレクサ437を制御して、差または減算前の値を選択してFCOUNTレジスタ431にセーブする。

【0233】キャッシュサービスのためにキャッシュバッファ312を使用するときは、同様な組の計算が必要である。マルチプレクサ433に対するモード信号がFCOUNTレジスタ431が計算のソースとして働くか、またはキャッシュバッファ312に記憶されていたバイト数を記憶する同様なCHCOUNTレジスタ432として働くかを選択する。

【0234】外部メモリ間転送を実行するとき、転送プロセッサ80は、リクエスト中のプロセッサのパラメータメモリの一部をバッファとして使用する。この外部メモリ間バッファのステートおよびその内容は管理が必要である。バッファポインタBUFPTRレジスタ414は、ソースマシン240と宛て先マシン340とによって共用されるが、その理由は、このバッファは1度に一つしかアクセスできないからである。BUFPTRレジスタ414は、ソースマシン320および宛て先マシン340の通常のロジックを使用して、バッファとの間で転送されたバイト数だけインクリメントされる。BUFPTRレジスタ414は長さが8ビットだけでよい。このアドレスの最高位ビットは、各プロセッサに対して固定されており、リクエスト中のプロセッサに基づき、一定のマルチプレクサから供給される。通常的一致/残留

スペースロジックを用いて、転送されたバイト数が計算される。外部メモリ間バッファを空にするソースマシン320は、BUFCOUNTレジスタ441を使用してそのバイトに残っている値を記憶する。外部メモリ間バッファを満たす宛て先マシン340は、FCOUNTレジスタ431を使用してバイトに残っている値を記憶する。

【0235】図13は、バッファカウント演算を示している。BUFCOUNTレジスタ441は、転送方向に応じて外部メモリ間バッファ内のスペアバイトの数または空にすべき残っているデータバイトの数の8ビットカウントである。ソースマシン320は、外部メモリ間バッファを満たしたとき、パケット転送用FIFOバッファ311にロードするバイト数だけBUFCOUNTレジスタ441をデクリメントする。BUFCOUNTレジスタ441が7以下になるとソースマシン320が停止し、パケット転送用FIFOバッファ311が外部メモリ間バッファ内に空にされるまで、宛て先マシン340の作動が続く。従ってソースマシン320は外部メモリ間バッファ内に入るデータよりも多いデータをフェッチできない。外部メモリ間バッファは、少なくとも120バイトで満杯になる。

【0236】外部メモリ間バッファが満杯になる状態と空になる状態の間で、BUFCOUNTレジスタ441は128から減算し、これを外部メモリ間バッファ内に残っているデータバイトの数にしなければならない。外部メモリ間バッファを空にすると、ソースマシン320はBUFCOUNTレジスタ441が0となるまで、これをデクリメントする。宛て先マシン340はパケット転送用FIFOバッファ311を空にし続け、再び方向を反転させる。外部メモリ間バッファの充填が開始する前に、BUFCOUNTレジスタ441は128にリセットされる。マルチプレクサ442および443、およびフル(全)加算器444は、このような必要な演算を行う。

【0237】パケット転送が一時中断されているとき、BUFCOUNTレジスタ441およびBUFPTRレジスタ414は、他のパケットパラメータがセーブされる。レストレーション時にオペレーションを正しく再開できるように、外部メモリ間バッファを空にするか、または満たす、現在の方向を表示するビットもセーブされる。

【0238】図14aおよび14bは、小エンディアンおよび大エンディアン係数のための、それぞれのポイントFDPTRおよびFSPTTRとパケット転送用FIFOバッファ311との関係を示している。パケットリクエストはソースおよび宛て先の形状を実質的に任意または従属的にできるので、宛て先に対して常に一致するわけではない。図8において、ソースと宛て先の整列がコンスタントに互いに変化した状態で、0バイトのサイク

ルと8バイトのサイクルとの間でソースマシン320は記憶し、宛て先マシン340はフェッチが可能である。従ってパケット転送用FIFOバッファ311は16バイトのバイト幅のFIFOであり、このFIFOはソースからの0~8バイトをロードできるのと同時に、宛て先への0~8バイトを空にできる。ソースマルチプレクサおよび整列ロジック330はソースから適当なバイトを抽出し、これらバイトをパケット転送用FIFOバッファ311の隣接するFIFOデータを維持する位置に記憶する。宛て先マルチプレクサおよび整列ロジック350は、パケット転送用FIFOバッファ311からの最も古いバイトを抽出し、宛て先に書き込む前に正しい位置に8バイトワードでこれらバイトを一致させる。ソースマルチプレクサおよび整列ロジック330、並びに宛て先マルチプレクサおよび整列ロジック350は、それらの転送を一致でき、できるだけ長くこの状態に留めることができるようになるとすぐに、64ビットのダブルワードの境界にこれらの転送を一致させる。従ってソースマルチプレクサおよび整列ロジック330並びに宛て先マルチプレクサおよび整列ロジック350は、変動する整列状態をサポートする。

【0239】パケット転送用FIFOバッファ311は、サーキュラバッファとして16ビットを取り扱う。データの開始点を任意のバイト位置とすることができ、宛て先マルチプレクサおよび整列ロジック350がデータを抽出する点は、4ビットのFDPTRレジスタによって表示され、FDPTRレジスタはデータを抽出する同じサイクルで変更され、ポインタを新しいデータの開始点に移動する。宛て先マルチプレクサおよび整列ロジック350は、抽出されたバイトを取り込み、宛て先ダブルワード内の正しい位置へこれらを整列させる。パケット転送用FDPTRバッファ311からの抽出および整列は宛て先マルチプレクサおよび整列ロジック350内の8つのパラレルな16ビットパレルシフトから成る16ポジションバイトローテータを使用した1工程で行われる。周辺のバイトをスワップするように、ガンのようなデータの操作を実行し、次に別の回転を行う。宛て先マルチプレクサおよび整列ロジック350も、宛て先に適当なバイトだけが書き込まれるように、同様にバイトストローブを操作する。

【0240】ソースマルチプレクサおよび整列ロジック330からのパケット転送用FIFOバッファ311へのローディングは、実質的に同様なオペレーションである。ソースマルチプレクサおよび整列ロジック330は、4ビットのFSPTTRレジスタによってマークされた最初の空いている位置にそのデータを書き込み、新しい最初の空いた位置を有するFSPTTRレジスタを更新する。パケット転送用FIFOバッファ311がフル状態のとき、FSPTTRレジスタは、一旦宛て先マルチプレクサおよび整列ロジック350があるデータを抽出す

れば、空となる最初の位置を実際にポイントする。

【0241】宛て先マシン340は、クロックサイクルのスレーブ位相に関するデータを抽出する。ソースマシン320はクロックサイクルの次のマスタ位相時にロードする。大エンディアンおよび小エンディアン作動モードをサポートするため、いずれの方向にもデータ回転を行う必要がある。データの連続性が常に正しくなるようにパケット転送用FIFOバッファ311の外側から見た場合、小エンディアンモードではデータ回転は時計回り方向にし、大エンディアンモードでは反時計回り方向にする。従って操作マルチプレクサおよび整列ロジック330、宛て先マルチプレクサおよび整列ロジック350、ポインタFSPTRおよびFDPTRおよびパケット転送用FIFOバッファ311のレジスタは、双方のエンディアンに合わせるように設計しなければならない。

【0242】図12のFCOUNTレジスタ431は、パケット転送用FIFOバッファ311にどれだけ多くのバイトが保持されているかの表示を記憶する。このFCOUNTレジスタ431は、17以上の値を決して含むことができない。このレジスタはこのレジスタ内に記憶された値がパイプライン内に現在あるインストラクションのすべてが完了した後に、パケット転送用FIFOバイト311内にどれだけ多くのバイトが入るかを表示するように、パイプラインのローディング前にソースマシン320および宛て先マシン340によって操作される。これによって、パケット転送用FIFOバッファ311のブロッキングを生じさせるようなパイプラインへのオペレーションのローディングができなくなる。従って、宛て先マシン340は常に十分なバイトを有し、ソースマシン320は常に十分な余裕を有することとなる。この結果、FDPTRまたはFSPTRが互いにオーバーテイクをする危険がなくなる。新しいパケットリクエストを開始すると、FDPTRおよびFSPTRが0にリセットされるので、これらの双方は同じ位置をポイントし、FCOUNTレジスタ431は0にリセットされる。これによりパケット転送用FIFOバッファ311の正しい初期化が保証される。

【0243】パケット転送が一時中断されると、FDPTR、FSPTRおよびFCOUNTレジスタ431のみならず、パケット転送用FIFOバッファ311レジスタ自身も制御される。これによりパケット転送用FIFOバッファ311の全ステートが保持され、よってこのステートはパケット転送の再開時にレストアできる。

【0244】値で満杯にされたパケット転送は、満杯にしている値をパケット転送用FIFOバッファ311の最下位のダブルワードにロードし、FCOUNTレジスタ431を16にセットし、FDPTRをエンディアンのための正しいデータ開始点にセットする。小エンディアンに対してはバイト0であり、大エンディアンに対し

てはバイト7である。値で満杯にされたパケット転送中では、FCOUNTレジスタ431の値は変更されない。従って宛て先マシン340は、満杯になったパケット転送用FIFOバイト311を常時見ているので、この満杯にしている値をフェッチし、整列する。ソースマシン320も満杯にされたパケット転送用FIFOバイト311を常時見ているので、必要に際し、停止する。値で満杯にされたパケット転送を一時中断またはレストアする際には、パケット転送FIFOバッファ311のステートは一時中断の別の形態に対する場合のように、セーブされ、レストアされるだけである。従って値で満杯にされたパケット転送を一時中断するためには特別な考慮は不要である。

【0245】ソースマシン320は、2つのパイプラインを含み、宛て先マシン340と別のパイプラインを共用する。パイプラインは内部メモリアンターフェース301および外部メモリアンターフェース302とインターフェースされ、4つの方向のデータ転送、すなわちオンチップからオンチップへ、オフチップからオンチップへ、オンチップからオフチップへ、オフチップからオフチップへのデータ転送を可能とする。ソースマシン320のアドレスパイプライン427およびバイトパイプライン428は、宛て先マシン340のパイプラインから分離されているが、共通のパイプラインの進行を行うように連動されている。これにより、ステップ状に維持しながら内部メモリアンターフェース310および外部メモリアンターフェース320上での独立した行動が可能となっている。あるインターフェースへの不連続性は、他のインターフェースを停止させることもある。アドレスパイプライン420はソースマシン320によってアクセスされるアドレスを記憶し、バイトパイプライン428はバイトストローブおよび整列情報を記憶する。この情報はどのバイトが対応しており、パケット転送用FIFOバッファ311またはキャッシュバッファ312にどれだけ多くのバイトをロードしたり、これらから除くかを表示している。転送コントローラ80のレジスタの転送にあたり、バイトパケット428はレジスタ番号を記憶する。サービスを受けた情報もバイトパイプラインを下方に移動し、オペレーションがいつ完了したかを表示する。

【0246】ソースマシン320と宛て先マシン340は、リクエストパイプライン429を共用する。リクエストパイプライン429はステータス情報、例えば低優先度パケット転送、ダイレクト外部アクセス等の実行中のオペレーションのタイプ、およびリクエスト中のプロセッサの表示を記憶する。リクエストパイプライン429は、ステータス事象が生じるときを表示するパラレル制御情報も記憶する。

【0247】一つ以上のアクセスが停止する際には、ときどき内部パイプラインにバブルが生じる。転送コント

ローラ80は、ガイドされた転送中にガイドテーブルの値をフェッチするのに、これらバブルを使用する。これにより、転送コントローラ80の転送中バンド幅をより効率的に使用できる。宛て先レジスタがフリーでない場合、このようなフェッチされたガイドテーブルの値は、GTP IPEレジスタ403の関連部分に記憶される。

【0248】転送コントローラ80はバイト配列されたパケット転送のみにあって、データ配列を行うことができる。ソースマシン320と宛て先マシン340とは独立しているので、これらマシンの各々は0バイトの各サイクルと8バイトの各サイクルとの間でフェッチしたり記憶することができる。このことは、ソースアドレスと宛て先アドレスとの相互の整列はコンスタントに変化し得ることを意味している。変動する整列をサポートするため、転送コントローラ80はパケット転送用FIFOバッファ311とソース配列ロジック332と、宛て先整列ロジック352を含んでいる。パケット転送FIFOバッファ311は、16バイトのFIFOレジスタであり、このレジスタはソースからの0～8バイトのコードと、宛て先への0～8バイトを空にすることを同時に行うことができる。ソースマルチプレクサ331およびソース配列ロジック332は、ソースから適当なバイトを抽出し、これらを先のソースバイトに隣接するパケット転送用FIFOバッファ311に記憶する。宛て先配列ロジック352および宛て先マルチプレクサ351は、パケット転送用FIFOバッファ311から最も古いバッファを抽出し、現在アドレス指定されている宛て先の8バイトのダブルワードで、正しい位置にこれらを配列する。パケット転送配列およびFIFO操作は自動的であり、意図する実施例ではプログラマーに対してトランスペアレントである。

【0249】キャッシュバッファ312にはパケット転送用FIFOバッファ311に対して作動が類似する8バイトのバッファであり、キャッシュおよびダイレクト外部アクセスオペレーション中に使用される。キャッシュの内外への転送は常に8バイト幅である。キャッシュバッファ312には外部メモリバスが60ビット幅よりも狭い場合にデータを配列するのを助ける。キャッシュバッファ312をパケット転送用FIFO場合に311と別個に設けたことにより、現在パケット転送用FIFOバッファ312にあるパケットデータを最初に空にすることなく、パケット転送の途中でより高い優先度のキャッシュおよびダイレクト外部アクセスリクエストをサービスすることが可能となっている。

【0250】転送コントローラ80はオンチップメモリにマッピングされる4つの、ユーザーがアクセス可能なレジスタを有する。これらレジスタはロードおよび記憶インストラクションによりマスタプロセッサ60によってアクセス可能である。これらレジスタはデジタル画像／グラフィックプロセッサ71、72、73、74に対

してはアクセスできない。表12はこれらレジスタをリストアップしたものである。

【0251】

【表12】

アドレス	TCレジスタ
0x01820000	REFCNTL
0x01820004	PTMIN
0x01820008	PTMAX
0x0182000C	FLTSTS

表12

【0252】REFCNTLレジスタは、システムのDRAMリフレッシュサイクルを制御するのに使用される2つの16ビットの値を含む。16ビットREFRATEフィールド（ビット15～0）は、DRAMリフレッシュサイクルを発生するインターバルを決定する。REFRATEにおける値は、各リフレッシュリクエストの間に生じるマルチプロセッサ集積回路100のクロックサイクル数を示す。REFRATEにおける32（16進数0020）未満の値は、DRAMリフレッシュをデイスエーブルする。REFRATEフィールドはリセット時に32（16進数0020）にセットされる。DRAMリフレッシュ中、16ビットの疑似アドレスが外部アドレスバス上に出力され、リフレッシュバンクデコーディングまたはRAS' だけのリフレッシュとともに使用される。16ビットRPARLDフィールド（ビット31～16）は、リフレッシュサイクル中に出力される最大値を含む。現在のリフレッシュアドレスのトラックを維持するのに、リフレッシュアドレスカウンタが使用される。このカウンタはリフレッシュサイクルが起きるたびにデクリメントされ、このカウンタが0に達すると、RPARLDにおける値が再ロードされる。RPARLDフィールドはリセット時に16進数0FFFFにセットされる。

【0253】パケット転送用最小レジスタPTMIN511およびパケット転送用最大レジスタPTMAX512は、パケット転送の作動時間を制御するようになっている。これらレジスタは、図31に示されており、以下、更に説明する。パケット転送用最小レジスタPTMIN511は、より高い優先度のパケット転送によってパケット転送をインタラプトできる前に、転送コントローラ80によってパケット転送のサービスを行わなければならないクロックサイクルの最小数を記憶する。好ましい実施例では、パケット転送用最小レジスタPTMIN511の24の最小位のビットしか実行されない。より高い位のビットからの読み出しは、常に0に復帰し、これらビットへの書き込みは全く効果がない。リセット時にパケット転送用最小レジスタPTMIN511には16進数10000（64Kサイクル）がロードされる。パケット転送用最大レジスタPTMAX512は、

タイムアウト前にパケット転送用最小レジスタPTMIN511により指定される時間の後にパケット転送を続けることができる最大時間を決定する。本実施例では、パケット転送用最大レジスタPTMAX512は24の最小位のビットを実行する。より高い位のビットからの読み出しは0に復帰し、これらビットへの書き込みは何らの効果もない。従ってPTMINサイクルの後により高い優先度のパケット転送によりパケット転送をインタラプトでき、PTMIN+PTMAXサイクルの後に同じ優先度の別のパケット転送を実行するため、このパケット転送を一時中断できる。より高い優先度のパケット転送のため、あるパケット転送を一時中断できるとき、パケット転送用最大レジスタPTMAX512内に記憶*

*された値は、影響しない。リセット時にPTMAXに16進数の値10000 (16Kサイクル) がロードされる。

【0254】パケット転送またはデジタル画像／グラフィックプロセッサ71、72、73、74のインストラクションキャッシュフィルサイクルまたはダイレクト外部アクセスサイクル中にフォールトが生じたことを表示するステータスビットを含む。表13は、FLTSTSレジスタ上の特定のビットの意味を示す。

【0255】

【表13】

FLTSTS ビット	名称	機能
31	DIGPC7	予約
30	DIGPC6	予約
29	DIGPC5	予約
28	DIGPC4	予約
27	DIGPC3	DIGP3キャッシュ／DEAフォールト
26	DIGPC2	DIGP2キャッシュ／DEAフォールト
25	DIGPC1	DIGP1キャッシュ／DEAフォールト
24	DIGPC0	DIGP0キャッシュ／DEAフォールト
23	DIGP7	予約
22	DIGP6	予約
21	DIGP5	予約
20	DIGP4	予約
19	DIGP3	DIGP3パケット転送フォールト
18	DIGP2	DIGP2パケット転送フォールト
17	DIGP1	DIGP1パケット転送フォールト
16	DIGP0	DIGP0パケット転送フォールト
0	M	MPパケット転送フォールト

表13

【0256】転送コントローラ80は、所定のメモリフォールトの検出に基づき、FLTSTSレジスタの個々のビットをセットする。マスタプロセッサ60のリクエストしたパケット転送中にフォールトが生じると、Mビット（ビット0）は1にセットされる。ビット19～16はデジタル画像／グラフィックプロセッサ71、72、73、74によってリクエストされるパケット転送フォールトを表示するのに使用される。DIGP3（ビット19）は、デジタル画像／グラフィックプロセッサ74のパケット転送フォールトを表示し、DIGP2（ビット18）はデジタル画像／グラフィックプロセッサ73のパケット転送フォールトを表示し、DIGP1（ビット17）はデジタル画像／グラフィックプロセッサ72のパケット転送フォールトを表示し、DIGP0（ビット16）はデジタル画像／グラフィックプロセッサ71のパケット転送フォールトを表示する。8つのデジタル画像／グラフィックプロセッサを含む実施例で、パケット転送フォールトをレポートするのにビット23～20が保留される。対応するデジタル画像／グラフィックプロセッサによりリクエストされるパケット転送中にフォールトが生じると、これらビットの各々が1にセ

ットされる。

【0257】DIGPC3ビット（ビット27）は、キャッシュフィルまたはデジタル画像／グラフィックプロセッサ74によりリクエストされるダイレクト外部アクセスオペレーション中にフォールトが生じると1にセットされ、同様に、DIGPC2ビット（ビット26）における1は、キャッシュフィルまたはデジタル画像／グラフィックプロセッサ73によりリクエストされるダイレクト外部アクセスオペレーション中のフォールトを表示する。DIGPC1ビット（ビット25）における1は、キャッシュフィルまたはデジタル画像／グラフィックプロセッサ72によりリクエストされるダイレクト外部アクセスオペレーション中のフォールトを表示する。DIGP0ビット（ビット24）における1は、キャッシュフィルまたはデジタル画像／グラフィックプロセッサ71によりリクエストされるダイレクト外部アクセスオペレーション中のフォールトを表示する。8つのデジタル画像／グラフィックプロセッサを使用する実施例では、付加的デジタル画像／グラフィックプロセッサによりリクエストされるキャッシュフィルまたはダイレクト外部アクセスオペレーションを表示するのに、ビット3

1～28が保留される。

【0258】M、DIGP3～0、またはDIGP3～0ビットのうちのいずれかをセットすると、マスタプロセッサ60がインタラプトされる。セットされているFLTSTSビットをクリアすると、関連するパケット転送、キャッシュフィルまたは外部アクセスが再スケジューリングされる。適当なビットに1を書き込むと、FLTSTSビットがクリアされ、1つのビットに0を書き込んで何の効果もない。

【0259】マスタプロセッサ60のレジスタは、スコアボード化されているので、フォールトを生じたダイレクト外部アクセスリクエストは必ずしもこのプロセッサのインストラクションパイプラインを停止するわけではない。マスタプロセッサ60はデータアクセスの別の試みを行う場合に限り、ダイレクト外部アクセス記憶装置のフォールトにより停止する。フォールトの生じたダイレクト外部アクセスロードは、同じ条件下またはダイレクト外部アクセスによりロードされるレジスタを使用する試みをなす場合に、マスタプロセッサ60を停止させる。デジタル画像／グラフィックプロセッサ71、72、73または74によりリクエストされるフォールトの生じたダイレクト外部アクセスは、リクエストされたダイレクト外部アクセスが完了するまでに、常にリクエスト中のプロセッサを停止させる。

【0260】パケット転送はメモリのうちの2つの領域間でのデータのブロック転送である。ソース(Src)メモリエリアから宛て先(Dst)メモリエリアまで、転送コントローラ80によってデータが転送される。ソースおよび宛て先エリアは、オンチップまたはオフチップメモリのいずれでもよい。パケット転送は、転送コントローラ80への領域としてマスタプロセッサ60またはデジタル画像／グラフィックプロセッサ71、72、73、74のうちの1つによって開始される。転送コントローラ80は、図6および7に示すように、リクエスト待ち行列化および優先度決定ロジック303により、固定された優先度決定方法およびラウンドロビン優先度決定方法を用いてリクエストをサービスする。一つのプロセッサがリクエストを送信すると、このプロセッサはプロセッサの実行を続けることができる。パケット転送はプロセッササイクルを追加することなく、転送コントローラ80によって完了される。パケット転送は、異なる優先度レベルに基づいて送信できるので、より高い優先度の転送は低い方の優先度の転送をインタラプトできる。このようなことが生じると、優先度の低い方の転送は転送コントローラ80によって一時中断され、転送内の現在位置が制御される。より高い優先度の転送が完了すると、インタラプトされたポイントで一時中断された転送が自動的に再開される。

【0261】パケット転送の理解を容易とするため、パケット転送を説明するのに用いる用語の簡単な定義を順

に説明する。ラインとはメモリにおける多数の隣接するバイトのことであり、パッチとは開始アドレスが等距離間隔しているライングループのことであり、パケットとはパッチの組であり、ピッチとは2つのライン間または2つのパッチの開始点の間のアドレスの差であり、パラメータテーブルとはデータのパケットおよびどのようにソースから宛て先に移動すべきかを記述するパラメータの、8ダブルワード長の組であり、リンクされたリストとはパラメータテーブルの組であり、テーブルの各々はリスト内の次のテーブルをポイントするものである。ガイドテーブルとは、パケット転送における個々のパッチを記述するパラメータのテーブルであり、ソース転送とは、ソースメモリ位置からのデータの転送のことであり、宛て先転送とは、宛て先メモリ位置へのデータの転送のことであり。

【0262】各プロセッサは関連するパラメータメモリを有し、転送コントローラがプロセッサからのパケット転送リクエストにサービスする際、転送コントローラ80が使用するため、関連するパラメータメモリ内に多数の位置が別個にセットされている。図15にはこれらエリアが示されている。転送コントローラ80により使用されるパラメータメモリエリアは、転送コントローラ80の使用のみに限定されているわけではない。しかしながら、これら位置に入れられるデータはパケット転送オペレーション中、転送コントローラ80によってオーバーライトされる可能性があることを知っておく必要がある。パケット転送リクエストがアクティブである時、これら位置に書き込みを行ってはならない。その理由は、このような書き込みによってパケット転送またはそのデータが破壊されるからである。

【0263】プロセッサがパケット転送を初期化するためのプロセス、方法またはシーケンスの一例は次のとおりである。対応するパラメータメモリ内にパケット転送パラメータテーブルを作成する。必要であれば、オンチップメモリ内にパケット転送用ガイドテーブルを作成する。ガイドテーブルの性質については後に更に説明する。第1パラメータテーブルの開始点をポイントするように、パラメータメモリ内のリンクされたリストの開始アドレスをセットする。図15は、このリンクされたリストの開始アドレスを記憶すべき対応するパラメータメモリ内の位置を示す。マスタプロセッサ60に対し、このリンクされたリストの開始アドレスは、16進数010100FCから16進数010100FFのアドレスに記憶しなければならない。デジタル画像／グラフィックプロセッサ71、72、73、74に対しては、このようなリンクされたリストの開始アドレスは16進数0100#0FC～16進数0100#0FF（ここで#はデジタル画像／グラフィックプロセッサの数である）のアドレスに記憶しなければならない。次に適当なパケット転送優先ビットおよびPビットをセットして、転送

コントローラ80にリクエストを送る。マスタプロセッサ60のためのPKTREQ制御レジスタ内およびデジタル画像／グラフィックプロセッサ71、72、73、74のCOMMレジスタ内に、これらビットが位置する。転送コントローラ80は、これらレジスタのステータスを検出し、図6および7に示された優先度に従ってパケット転送リクエストのサービスをする。

【0264】パケット転送リクエストはリンクされたリスト構造のように送られる。リンクされたリストは単にパケット転送パラメータテーブルの集合であり、ここでは各パケット転送は、リスト上の次の入力の入力ポイントに対するポインタを含む。パケット転送はオンチップまたはオフチップメモリ上で作動できるが、パラメータテーブル自体のリンクされたリストはオンチップメモリに記憶しなければならない。各プロセッサはメモリに記憶された多数のリンクされたリストを有することができるが、一時にそのうちの一つしかアクティブになれない。リクエスト中のプロセッサのパラメータメモリ内の専用のリンクされたリストの開始アドレス位置に、アクティブなリンクされたリストの開始点が記憶される。リンクされたリスト内の各入力パラメータテーブルは、リスト上の次の入力の位置に対するポインタを含む。リストの終了部はパケット転送パラメータテーブルのパケット転送オプションフィールド内のストップビットによってマークされているので、リスト内の最終入力は、いずれの位置もポイントできる。

【0265】図16は、簡単なリンクされたリスト構造を示す。このリストは2つのパケット転送パラメータテーブルを含む。リスト開始ポインタは、第1パケット転送パラメータテーブルの開始アドレスを記憶し、第1パケット転送パラメータテーブル内の第1データは、第2の、すなわち本例では最終のパケット転送パラメータテーブルのアドレス内にある。本例は、2つのパケット転送パラメータテーブルしか含まないが、リンクされたリストは、オンチップメモリに合わせるよう、できるだけ長くすることができる。

【0266】あるプロセッサがPビットをセットすることによりパケット転送を送った場合、転送コントローラ80は対応するプロセッサのQビットをセットすることによって応答する。このことは、そのプロセッサのリンクされたリストがパケットコントローラ80内に待ち行列状になっていることを表示する。パケット転送優先レベルに適当なラウンドロビントークンがリクエスト中のプロセッサに達すると、転送コントローラ80はリクエストのサービスをアクティブに開始する。転送コントローラ80がリクエスト中のプロセッサのパラメータメモリ内のリンクされたリストの開始位置を読み出す。転送コントローラ80は第1パケット転送パラメータテーブルの内容を、そのソースマシン320および宛て先マシン340内のレジスタに読み出す。次に転送コントロー

ラ80はソースマシン320および宛て先マシン340を使用して、パラメータテーブル内に表示されたようにデータを転送する。パケット転送が完了すると、転送コントローラ80はリンクされたリスト内の次のアドレスと共に、パラメータメモリ内のリンクされたリストの開始アドレスを更新する。このアドレスは、ちょうど完了したパケット転送パラメータテーブルの第1入力から読み出される。転送コントローラは次に、リンクされたリスト上の最後の入力に達するまで、この手順を繰り返す。

【0267】本明細書に述べたパケット転送デバイス、プロセスおよび方法は、データの移動のフレキシビリティを最大にできるよう、多数の異なるフォーマットおよびオプションを提供するものである。これらフォーマットを検討する際、パケット転送のソース転送と宛て先転送とは互いに独立していることに留意することが重要である。これにより読み出し時のフォーマットと完全に異なるフォーマットを利用してパケットデータを書き込むことができ、これにより任意の数のスプレッド機能またはマージ機能を自動的に達成できる。これら2つの基本的パケット転送フォーマットは、次元が決定され、ガイドされている。これらフォーマットはソース転送またはソース転送を記述しているかまたは宛て先転送を記述しているかに応じて、どのようにデータを読み出すか、または書き込むかを決定できる。ソース転送と宛て先転送に対して異なるフォーマットを規定することが可能であることに留意されたい。

【0268】次元の定められた転送は、最もシンプルなタイプであるが、最も硬直したタイプでもある。このような次元の定められた転送はデータバイトの簡単な隣接するリニアシーケンスでもよいし、または多数のかかる領域から構成できるソースまたは宛て先を記述する。アドレス指定機構は三次元までのアレイの指定を可能とする。この機構は、単一パケット転送による多数の二次元状パッチの転送を可能とする。第1またはA次元に沿うデータは、常に1バイトだけ離れている。第2すなわちB次元および第3すなわちC次元に沿う間隔は任意であるが、パッチ全体に対して固定されている。宛て先次元が完了すると、転送が完了する。

【0269】図17は、次元の定められた転送がどのようにソースメモリまたは宛て先メモリにアクセスするかを示している。この図は3ラインの2パッチから成るパケットを示しており、各パッチは512個の隣接する8ビットピクセルから成る。例えば2つのデジタル画像／グラフィックプロセッサ71、72、73、74が3×3の畳み込み（各々はラインのパッチの1つに作用する）を実行しようとしている場合、これが必要となり得る。第1パッチ（PQR）は、デジタル画像／グラフィックプロセッサ71と関連するデータメモリ22に転送すべきデータを示し、第2パッチ（STU）データは

デジタル画像／グラフィックプロセッサ72に関連するデータメモリ27に転送すべきデータを示す。本例はソースメモリエリアから転送コントローラ80がデータをどのように読み出すべきかを指定しているの、ソース転送を示していることになる。データパケットは、次のパラメータ、すなわちAカウント、Bカウント、Cカウント、開始アドレス、BピッチおよびCピッチによって特定される。Aカウントとは、第1次元内の隣接するデータバイトの数であり、図17の例ではこの数は512である。Bカウントとは、パッチを形成する工程の数または1未満のライン数である。図12はBカウントが2である場合を示す。Cカウントは1未満のパケットを形成するためのパッチ工程の数である。図17はCカウント数が1である場合を示す。開始アドレスとは、パケットの開始点のリニアアドレスである。このアドレスはPと表示されるバイトのアドレスである。Bピッチとは第2次元のリニアピッチのことである。すなわちバイトPとQとのアドレスの差またはバイトQとRとのアドレスの差である。Cピッチとは第3次元のリニアピッチ、すなわちバイトPとSとのアドレスの差のことである。ソース転送および宛て先転送のいずれも、このように定義できるが、ソースと宛て先の形状を全く異なるようにできるように、パラメータは独立したものである。

【0270】次元の定められた転送のすべての次元をアクティブにする必要はない。Bカウントおよび／またはCカウントを0にセットすることにより、転送を個々のバイト、ピクセルの倍数バイト、ラインまたはパッチに限定できる。0のAカウント値は転送されるデータを0にするので、エラーを発生し得る。

【0271】ガイド転送とは次元アドレスのシーケンスをパケット転送パラメータ内の値のみから計算するのではなく、オンチップメモリ内のテーブルからガイドする転送のことである。これらのオペレーションは次元の定められた転送よりも複雑であるが、よりフレキシブルである。ガイド転送には2つのクラス、すなわち固定パッチ転送と可変パッチ転送とがある。固定パッチガイド転送は次元の定められた転送の場合のようにパケット転送パラメータ内に記述された第1次元および第2次元を有するが、第3次元はオンチップガイドテーブル内のエントリからガイドされる。可変パッチガイドテーブルではガイドテーブルは各パッチに対しA次元およびB次元のサイズも決定する。いずれの形態でも、2つの1次元は次元の定められたテーブルと同じようにアクティブである。従って各ガイドテーブルエントリが2つの第1次元のサイズに従って個々のバイト、倍数バイトの個々のピクセル、ラインまたは2次元パッチを移動できる。これによりルックアップテーブルによるライン描きまたはデータ処理の際に生じるような多数のイレギュラーなオペレーションが可能となる。

【0272】特に表示しない限り、ガイド転送について

の次の記載における情報は、ソース転送および宛て先転送の双方にも同じように当てはまる。ガイドテーブルは単なるエントリのブロックであり、これは転送のタイプに応じて32ビットまたは64ビットとなり得る。このガイドテーブルは、オンチップメモリ内に位置していなければならない。ガイドテーブルは固定パッチ転送に対しては、32ビットワードのアドレスに整列していなければならない、可変パッチ転送に対しては64ビットのダブルワードアドレスに整列していなければならない。ガイドテーブルが含むガイドテーブルの開始アドレスおよびエントリの数は、対応するパケット転送パラメータテーブル内に表示される。各ガイドテーブルエントリはパケット転送内の2次元パッチに対応しており、転送コントローラ80がパケット転送サービスをする際、このコントローラはパケット転送内の次のパッチの処理をするのに必要のように、ガイドテーブルエントリを一つずつフェッチする。ガイドテーブルエントリの詳細については、下記のガイド転送の詳細な説明で述べる。

【0273】固定パッチガイド転送は、32ビットエントリを含むオンチップガイドテーブルを使用する。このテーブルは、ワード整列されていなければならないので、テーブルのエントリアドレスの最小位の2つのビットは00でなければならない。各エントリは3次元の転送のためのアドレスを計算するのに使用される情報を含む。固定パッチ転送は3つのタイプがある。すなわち固定パッチデルタガイド転送、固定パッチオフセット転送および固定パッチオフセットガイドルックアップテーブル転送がある。これらタイプについては、下記により詳細に説明する。

【0274】固定パッチデルタガイド転送に対し、ガイドテーブルは現在のパッチの開始アドレスを形成するのに、先の2次元パッチの開始アドレスに追加すべき32ビットのデルタ値を含む。パッチサイズはAカウントおよびBカウントパケット転送パラメータによって固定され、定義される。

【0275】図18に、固定パッチデルタガイドパケット転送のプロセスおよび方法の一例が示されている。ここで、第1パッチすなわちパッチAの開始アドレスを形成するため、パケット転送パラメータ内に示された開始アドレスに値デルタAが加算される。この開始アドレスは最終パッチ開始レジスタ403に記憶され、パッチBの開始アドレスを形成するのに最終パッチ開始点403に記憶されたパッチAの開始アドレスに、デルタBが加算され、同様に、その後同じような加算が行われる。ここで加算値は次のパッチの開始アドレスのベースを形成するように、最終パッチ開始レジスタ403に常時記憶されることに留意されたい。図18に示すように、テーブルポインタ501は、ガイドテーブル502内の次のエントリをポイントするように、各パッチの後に4バイ

トだけインクリメントされる。多数のエントリーカウンタ503には、最初パケット転送パラメータテーブルからのエントリー値の数がロードされており、エントリーカウンタ503の数は各パッチの後で一つだけデクリメントされる。エントリー各503の数が0に達するとパケット転送が終了する。

【0276】固定パッチオフセットガイドパケット転送は、ガイドテーブル502を使用し、このテーブル502は各パッチの開始アドレスを形成するため、パケット転送パラメータ内に示されるベースアドレスに加算すべき32ビットの値を含む。パッチサイズはAカウントおよびCカウントパケット転送パラメータにより固定され、定義される。

【0277】図19は固定パッチオフセットガイドパケット転送のためのアドレス指定機構プロセスおよび方法を示す。ガイドテーブル502における第1エントリーからの値デルタAは、パッチAの開始アドレスを形成するため、パケット転送パラメータテーブルに特定され、ベースアドレスレジスタ505に記憶されたベースアドレスに加算される。次に、パッチBの開始アドレスを形成するよう、ベースアドレスレジスタ505内に記憶されたアドレスにデルタBが加算され、ガイドテーブル502内の最終エントリーまでに同様な加算が行われる。ここで、パケット転送パラメータテーブル内に特定されたベースアドレスが0になると、ガイドテーブル502は絶対アドレスを指定する。デルタガイド転送の場合と同じように、テーブルポインタ501は現在のガイドテーブルエントリーをポイントし、エントリーカウンタ503の数は実行するパッチの数をトラッキングする。

【0278】ガイドテーブル502は、固定パッチオフセットガイドルックアップテーブルパケット転送に対し、32ビットのオフセット値を含む。この値はオフセットレジスタ506内で0フィルの状態では0、1、2または3ビットだけ左にシフトし、ベースアドレスレジスタ505内に記憶されたアドレスに加算されるべきものである。ベースアドレスレジスタ505に記憶されたベースアドレスは、パケット転送パラメータに示されており、これによりルックアップテーブルのデータサイズと独立したルックアップテーブルのオペレーションのために転送を利用できる。固定パッチオフセットガイドルックアップテーブルフォーマットは、ソース転送に対して使用できるだけである。シフト量はパケット転送パラメータのAカウントフィールドのビット1~3のうちの最も左側のビットの位置によって表示されている。ビット3のうちの1は、左にシフトされた3つの場所を示し、ビット2は2つの場所を示し、ビット1は1つの場所を示す。ビット1~3の中に1がない場合は、0シフトを示す。このような左シフトにより、8、16、32および64ビットのサポートが可能となる。ルックアップテーブル転送のためのパッチサイズは、固定された1次元

であり、1、2、4または8バイトにセットされる。

【0279】図20は、固定パッチオフセットガイドルックアップテーブルパケット転送のためのアドレス計算の一例を示す。ここでオフセットAはAカウントフィールド内の値によって示されるようなルックアップテーブルのデータサイズに従って、0、1、2または3ビットだけ左にシフトされる。オフセットレジスタ505内のこのようなシフトされたアドレスは、パッチAの開始アドレスを形成するため、ベースアドレスレジスタ505内に記憶されたベースアドレスに加算される。次に、オフセットB値がシフトされ、ベースアドレスレジスタ505内のベースアドレスに加算され、パッチBの開始アドレスを形成し、同様な操作が次々に実行される。ガイドテーブル502からオフセット値がロードされる際にシフトが行われる。先の実施例で述べたように、テーブルポインタ501およびエントリーレジスタ503の数は、ガイドテーブル502内の現在位置およびパッチの数のトラッキングを維持する。

【0280】可変パッチガイド転送は、パケット転送パラメータ内でなく、ガイドテーブル内のすべてのパッチサイズ情報を特定する。これによりパケット転送内の各パッチが異なる次元を有することができる。転送はデルタガイドまたはオフセットガイドのいずれでもよい。ガイドテーブルは可変ガイドパッチ転送に対し64ビットのダブルワードエントリーから成る。図21にガイドテーブル用の小エンディアンフォーマットが示されている。ダブルワードの下方の半分は第1の2つの次元に対するAカウント値およびBカウント値を含む。上方の32ビットは第3次元のアドレスを計算するのに使用される値を含む。図22は、大エンディアンガイドテーブルフォーマットを示し、このフォーマットでは、上方の32ビットがAカウント値およびBカウント値を含み、下方の32ビットがオフセットまたはデルタを含む。

【0281】32ビットの書き込みを用いることにより、エンディアンフォーマットから独立したガイドテーブルを作成するためのソフトウェアが適当に与えられる。AおよびBカウントはワード1のアドレスに現れ、オフセット/デルタアドレスはワード0のアドレスに現れる。ここでワード1はワード0よりも大きいアドレスの4バイトにある。転送コントローラ80は64ビットのアクセス中に一時に双方の32ビットワードをアクセスし、そのとき選択されているエンディアンモードに従って正しい内部オペレーションをするのに必要なワード順序を調節する。可変パッチ転送のためのガイドテーブルエントリーはダブルワードに整列すべきである。すなわちバイトアドレスの3つの最小位ビットを000としなければならない。

【0282】図23は、可変パッチのデルタガイドパケット転送の一例を示す。可変パッチデルタガイドパケット転送に対し、ガイドテーブルエントリーの第1ワード

は、最後のパッチ開始レジスタ504内に記憶された先のパッチの開始アドレスに加算すべき32ビットのデルタ量を含む。パケット転送パラメータには第1パッチのための開始アドレスが示されている。パッチサイズは可変であり、各ガイドテーブルエントリーの第2ワード内に指定される。ここで、パッチWの開始アドレスを形成するのに、パケット転送パラメータ内に特定された開始アドレスにデルタWが加算される。Aカウンタは第1次元のバイトの数の長さを決定し、Bカウンタは第2次元のサイズ（これはライン-1の数である）を決定する。次にパッチXの開始アドレスを発生するよう、最終パッチ開始レジスタ504内に記憶されたパッチW開始アドレスにデルタXが加算される。先に固定パッチ転送の場合に述べたように、テーブルポインタ501およびエントリーカウンタ503の数は、ガイドテーブル502内の位置のトラッキングを維持する。各ガイドテーブルエントリーは、64ビットすなわち8バイトであるので、テーブルポインタ501は、8バイトよりも大きいアドレスをポイントするようにインクリメントされる。

【0283】図24には、可変パッチオフセットガイドパケット転送の一例が示されている。可変パッチオフセットガイドパケット転送のための各ガイドテーブルエントリーの1ワードは、32ビットオフセット値を含む。この32ビットのオフセット値は、各パッチの開始アドレスを計算するよう、ベースアドレスレジスタ505内に記憶されたベースアドレスに加算される。このベースアドレスはパケット転送パラメータ内で特定される。パッチサイズは可変であり、各ガイドテーブルエントリーの他方の半分に特定される。パッチWの開始アドレスを発生するよう、ベースアドレスレジスタ505内に記憶されているベースアドレスに、アドレスWの値が加算される。Aカウンタ値およびBカウンタ値はパッチのサイズを決定する。パッチXのための開始アドレスを得るよう、ベースアドレスレジスタ505内に記憶された下のベースアドレスにオフセットXが加算される。テーブルポインタ501およびエントリーカウンタ503の数は、ガイドテーブル502内の位置のトラッキングを維持しながらテーブルポインタ501が8だけインクリメントされる。

【0284】ルックアップテーブルパケット転送に類似する、値でフィルする（fill-with-value）パケット転送は、ソース転送に対して指定できるにすぎない。この値でフィルするパケット転送は実際にはソースメモリからデータを転送するのではなく、むしろパケット転送パラメータ内にソースの値を特定するようになっている。2つの32ビットフィールド、最大位のフィル値のワードおよび最小位のフィル値のワードが、宛て先メモリを満たすのに使用される64ビットの値を特定する。フィルパターンが64ビット未満であれば、最小値のフィル値のワードおよび最大位のフィル値のワ

ードにわたって、このパターンを再現しなければならない。フィル値では整列オペレーションが実行されず、宛て先ダブルワードに書き込まれるバイトはフィル値のダブルワードからの対応するバイトとなっている。値でフィルするパケット転送に対して、ソース開始アドレスまたはソース次元カウンタは特定されない。パケット転送のサイズは宛て先転送パラメータによって決定される。

【0285】種々の転送フォーマットによりソース転送と宛て先転送の多数の組み合わせが可能となっている。これら組み合わせを特定するため、パケット転送パラメータテーブル内のパラメータのフォーマットは、必要なソース転送および宛て先転送のタイプに応じて変わる。パラメータテーブルの内容の説明を補助するため、図25～29に種々の可能なパラメータの例が示されている。

【0286】パケット転送パラメータテーブルは、オンチップメモリすなわちマスタプロセッサ60のパラメータメモリ15、デジタル画像／グラフィックプロセッサ71、72、73、74のパラメータメモリ25、30、35または40、もしくは、デジタル画像／グラフィックプロセッサ71、72、73、74のデータメモリ22、23、24、27、28、29、32、33、34、37、38または39内にあることが好ましい。パケット転送パラメータテーブルは、整列された64バイトでなければならない。すなわち6つの最小位アドレスビットが000000となる。どのメモリがこのテーブルを含むかについては、制限はない。従ってマスタプロセッサ60はデジタル画像／グラフィックプロセッサ71に対応するパラメータメモリ25にあるパラメータテーブルを使用することができる。リクエスト中のプロセッサはパケット転送リクエストを送る前に自己のパラメータメモリ内のリンクされたリストの開始アドレス位置に適切な開始アドレスを入れるだけである。

【0287】図25～28に関連して、次の章はパケット転送パラメータ内の種々のフィールドについて述べる。PTによって表示されるパケット転送パラメータの開始アドレスに対する各フィールドのアドレスは、フィールドが有効な転送タイプと同じように示される。多くの場合、2つの同じフィールドがあり、1つはソース転送用であり、他方は宛て先転送用である。これらの場合、1つのコンテキストでフィールドの記述が示される。あるフィールドは、転送のタイプに応じて使用しないままにできる。表14は可能なオプションのすべてをリストアップしたものである。以下、これらのオプションについて詳細に説明する。

【0288】

【表14】

オペレーション	転送元	転送先
次元の定められた転送	y e s	y e s
固定パッチデルタガイド転送	y e s	y e s
固定パッチオフセットガイド転送	y e s	y e s
固定パッチオフセットガイドしU T	y e s	n o
可変パッチデルタガイド転送	y e s	y e s
可変パッチオフセットガイド転送	y e s	y e s
値で満たす転送	y e s	n o
トランスベアレントの次元の定められた転送	y e s	n o
トランスベアレント固定パッチデルタガイド転送	y e s	n o
トランスベアレント固定パッチオフセットガイド転送	y e s	n o
トランスベアレント可変パッチデルタガイド転送	y e s	n o
トランスベアレント可変パッチオフセットガイド転送	y e s	n o
ブロック書き込み	n o	y e s

表14

【0289】パケット転送パラメータ内の第1データワードは、次のパラメータテーブルの次のパケット転送開始アドレスである。このワードはすべてのパケット転送タイプに存在し、パケット転送用のリンクされたリスト上の次のエントリーの開始点に対する32ビットのポインタである。このワードは64バイトの整列したオンチップアドレスをポイントするので、このアドレスの6つの最小位のビットは000000とならなければならない。パケット転送用オプションフィールド内の停止ビットは、リンクされたリストを終了させるのに使用されるので、リンクされたリストの最終エントリーは、次の特別なアドレスエントリーを必要としない。パケット転送が成功裏に終了するときはいつも、リクエスト中のプロセッサのパラメータメモリ内のリンクされたリストの開始アドレス位置に、次のパケット転送エントリーアドレスワード内の値が書き込まれる。これによりポインタは次のパケット転送に自動的に進む。このポインタは、リンクされたリストに実行を休止するのに停止ビットが使用される場合に、このポインタがリンクされたリストがイネーブルされた際にリンクされたリスト上の次のパケット転送をポイントするように、リンクされたリスト上の最終パケットが完了した後にも、更新される。

【0290】第2ワードは、パケット転送オプションフィールドとなっている。これはPT+4に位置する。このパケット転送オプションフィールドは、データを転送する方法で種々のオプションを指定するようにすべてのパケット転送タイプで使用される。以下、このパケット転送オプションフィールドについて詳細に説明する。

【0291】次元が定められたパケット転送およびデルタガイドパケット転送で使用される第3および第4データワードは、ソースおよび宛て先開始アドレスとなっている。PT+8にはソース開始アドレスがあり、PT+12には宛て先アドレスがある。これらフィールドは次元の定められたパケット転送およびデルタガイドパケット転送で使用される。次元の定められたパケット転送に対し、各々はソースまたは宛て先のための開始バイトアドレスを表示する32ビットのワードであり、デルタガ

イド転送に対し、各々はソースまたは宛て先転送のために第1デルタオフセット値を加算する開始アドレスを示す。

【0292】オフセットガイドパケット転送で使用される第3および第4データワードは、PT+8にはソース開始アドレスがあり、PT+12には宛て先アドレスがある。これらフィールドは、オフセットガイド転送モードを使用する際に、ソースおよび宛て先開始アドレスの場所をとる32ビットのオフセット値である。

【0293】第5および第6データワードは、ソースおよび宛て先のAおよびBカウントであり、PT+16バイトにおける第5データワードはビット15~0にソース用のAカウントを有し、ビット31~16にソース用のBカウントを有する。同様に、PT+20における第6データワードは、ビット15~0に宛て先用Aカウントを有し、ビット31~16に宛て先用Bカウントを有する。このAカウントの16ビットフィールドは、次元の決められた、すなわち固定パッチ転送のためのソースまたは宛て先の対応する第1次元において、転送すべきバイトの数を指定する。可変パッチガイド転送に関しては、Aカウントフィールドは使用されない。更に値でフィルされたパケット転送に関してはソースAカウントは使用されない。Bカウントの16ビットフィールドは、次元の定められた、または固定されたパッチ転送用のソースまたは宛て先の対応する第2の次元で発生すべきステップ数を特定する。これはライン-1の数に等しい。従って0の値は第2の次元をディスエーブルし、その結果パッチ当たり1本のラインしか転送しない。ソースBカウントは値でフィルされたパケット転送中には使用されない。

【0294】次元の定められたパケット転送内のPT+24およびPT+28に位置する第7および第8データワードは、それぞれソースCカウントおよび宛て先Cカウントである。これら32ビットのフィールドはソースまたは宛て先の第3次元におけるパッチ工程数を特定する。従って0の値は第3次元をディスエーブルし、一つのパッチしか転送しない。フィル転送に対してはソース

Cカウントフィールドを使用されない。

【0295】デルタまたはオフセットガイド packets 転送における第7および第8データワードはエントリーの数である。これらの32ビットフィールドにおける値は、それぞれソースおよび宛て先のためのガイドテーブルのエントリーの数を指定する。従ってこれらフィールドは、転送される情報のパッチ数を表示する。値が0であればデータ転送は行われない。

【0296】PT+32およびPT+36には第9および第10データワードがある。次元の定められた固定パッチ packets 転送では、これら32ビットのフィールドはソースおよび宛て先の第2次元のピッチを特定する。このピッチの値はソースまたは宛て先ラインの開始アドレスに加算され、次のラインの開始アドレスを得る。対応するBカウントフィールドにおける値は、0であるが、このフィールドはプログラムされない状態のままにできる。

【0297】PT+40およびPT+44にはそれぞれ第11および第12データワードがある。次元の定められた packets 転送では、これら32ビットフィールドでは、ソースまたは宛て先の第3次元のピッチを特定する。次のパッチの開始アドレスを得るのに、ソースまたは宛て先パッチの開始アドレスにこのピッチの値が加算される。対応するCカウントフィールド内の値が0であれば、このフィールドはプログラムされない状態のままにできる。

【0298】デルタまたはオフセットガイド転送において、第11および第12データワードは、ガイドテーブルアドレスである。これらフィールドにはそれぞれのガイドテーブル内の第1エントリーをポイントする整列されたアドレスがロードされる。これらデータワードの値は、ガイドテーブルからエントリーが取り出されるたびに固定パッチ転送に対しては4だけ、または可変パッチ転送に対しては8だけ自動インクリメントされる。これらガイドテーブル開始アドレスは、固定パッチ packets 転送に対しては32ビットワードアドレスに対して整列し、可変パッチ packets 転送に対しては64ビットダブルワードアドレスに整列しなければならない。ガイドテーブルはオンチップメモリ内になければならず、いずれかの条件が真でなければ、イリーガルなアドレスエラーインタラプトが生じる。

【0299】第9および第11データワードは、値で満たされた packets 転送において特別な用途がある。これら2つの32ビットフィールドは64ビットフィルパターンを構成する。宛て先のダブルワードに書き込まれたバイトはフィル値のダブルワード内の対応するバイトから取り出される。従ってフィルパターンが64ビット未満であれば、これら2つのデータワード内でデータ値を繰り返さなければならない。値でフィルする packets 転送に対し、このフィル値はソースに対してのみ可能であ

るが、他方、宛て先は次元の定められた、固定または可変パッチのデルタまたはオフセットガイドにできる。

【0300】 packets 転送がソーストランスペアレンシーを利用する場合、第13および第14データワードはトランスペアレンシー値をホールドする。この64ビットのダブルワードはPT+48で開始する。この64ビットフィールドはトランスペアレンシーを選択した場合、比較すべき値を表示するのに使用される。 packets 転送オプションフィールドの packets アクセスモードフィールドは、このフィールド内に含まれる値のトランスペアレンシーサイズまたは数を表示する。宛て先ダブルワードにすぐに書き込まれるバイトは、トランスペアレンシー値内の対応するバイトと比較され、一致が見つかればバイトストローブが非アクティブにドライブされる。ピクセルサイズが64ビット未満であれば、データ値を再現しなければならない。このフィールドはブロックの書き込みを除くすべての非トランスペアレンシー転送に対しては用いられない。ソーストランスペアレンシー値は、エンディアンモードにかかわらず、メモリに書き込まれるのとまさに同じように、転送コントローラ80によって使用されるので、ワードスワップは生じない。これは図25に示されている。 packets 転送パラメータテーブル開始アドレスに続く48バイトをアドレス指定するダブルワード書き込みとして常にこの値を書き込むことにより、混乱を避けることができる。

【0301】 packets 転送はブロック書き込みモードを使用する場合、第3および第4データワードはカラーレジスタのデータをホールドする。64ビットのカラーレジスタフィールドはブロック書き込みサイクルの準備の際にビデオランダムアクセスメモリ (VRAM) をロードするのに使用される値を含む。この値は、シミュレートされたブロック書き込みを実行する際にも使用される。転送コントローラ80はエンディアンモードにかかわらず、トランスペアレンシー値に関してこれまで述べたのとまったく同じ態様で書き込まれるカラーレジスタ値を使用する。これは図29に示されている。以下、このフィールドの使用について詳細に説明する。

【0302】 packets 転送パラメータの最終ダブルワードを形成するPT+56で開始する第15および第16データワードは、すべての転送モードに対してそのときは使用されず、プログラムされない状態のままにできる。これらデータワードはマルチプロセッサ集積回路100を未来バージョンで使用できる。

【0303】図25～29は、 packets 転送パラメータテーブルの少数の可能なフォーマットを示す。図25は、ソーストランスペアレンシーを有するソースおよび宛て先用の次元の定められた packets 転送用のパラメータのテーブルの一例を示し、PT+48にトランスペアレンシーワード0が記憶され、PT+52にトランスペアレンシーワード1が記憶される。図26は、値でフィ

ルする宛て先用の次元の定められたパケット転送用パラメータテーブルの一例を示す。PT+40にフィル値の最大のビットが記憶される。PT+32にフィル値の最小位のビットが記憶される。図27は、次元の定められたソースおよび固定パッチガイド宛て先パケット転送用のパラメータテーブルの一例を示す。PT+12におけるデータワードは、ベースアドレスを記憶し、PT+28におけるデータワードはガイドテーブル内のエントリ数を記憶し、PT+44におけるデータワードはガイドテーブルの開始アドレスを記憶する。図28は、次元の定められたソースおよび可変パッチガイド宛て先パケットのためのパラメータテーブルの一例を示す。通常PT+22に記憶される宛て先AおよびBカウントはプログラムされないが、その理由は、これらカウントが宛て先ガイドテーブルの一部であるからである。図29は、次元の定められたブロック書き込みパケット転送用の一例を示す。PT+48で開始するダブルワードは、カラーレジスタワード0および1を含むことに留意された。

【0304】パケット転送オプションフィールドはソースおよび宛て先転送のためにどんなフォームの転送を用いるかを選択し、現在のパケットがリンクされたリストを終了させるかどうかを決定する。更に特殊転送モード、パケット転送終了時の付加的地址計算の実行、次元アドレス計算の方向変化またはソース転送と宛て先転送の反転のような多数の付加的特徴の選択を可能にする*

ビット		転送先更新オペレーション
1	0	
0	0	更新せず
0	1	宛て先転送内の最終ラインの開始アドレスにBピッチを加算（これより減算）し、結果を転送先に書き込む。
1	0	宛て先転送内の最終パッチの開始アドレスにCピッチを加算（これより減算）し、結果を転送先に書き込む。
1	1	宛て先転送内の最終パッチの開始アドレスにCピッチを加算（これより減算）し、その結果をPTパラメータテーブル内の転送先および開始アドレスに書き込み、次にPTオプションフィールド内の反転転送先Cアドレス指定ビットをトグルする。

表15

【0308】宛て先更新モードは、使用中の最大次元の追加ステップを実行するのに使用できるが、他の有効なオペレーションも実行できる。転送は2次元だけに行うことができたが、Cピッチを加えるよう、モード10も使用できる。これにより最終パケットに対し次の2次元パケットを位置させることができる。2つの1次元または2次元パッチ間で、例えばオンチップデータメモリ上でピンポン動作するパケット転送を再送信するのにモード11が特に有効である。完了時にパケット転送パラメータを更新するたびに、第3次元のアドレス指定方向が

*る。図30にパケット転送オプションフィールドのフォーマットが示されている。オプションフィールドにすべて0をロードすることにより、特殊アドレスモードを用いないで、ソースおよび宛て先で次元の定められた転送を利用するデフォルトパケット転送を行うことになる。

【0305】パケット転送オプションフィールドのビット1~0は、宛て先更新モードフィールドを形成する。これら2つのビットは、元のパケット転送パラメータテーブル内の宛て先開始アドレスが、パケット転送が完了した場合にどのように更新されるかを表示する。これらビットが0でなければ、パケット転送完了後に余分なアドレス計算を行う。パケット転送パラメータテーブルに指定された元の宛て先開始アドレス上にこの値が書き込まれる。これによりパケット転送を更に送ることが可能となり、オフ状態から続けることが可能となる。これは特に2つのメモリエリア間でピンポン操作するのに有効である。

【0306】表15に、宛て先更新モードがリストアップされている。下記のような宛て先反転アドレス指定ビットの反転宛て先Cまたは反転宛て先Bの一方がセットされると、加算よりも減算を行う。

【0307】

【表15】

反転されるので、この方向はメモリエリア間で交互に変わる。

【0309】この機能は、主に次元の定められた転送と共に使用することを意図するものである。この機能は、ガイド転送で指定できるが、注意が必要である。ガイド転送に対しては、宛て先Cピッチは宛て先ガイドテーブルポイントと置換されていることを思い出していただきたい。

【0310】パケット転送オプションフィールドのビット6~4は、宛て先転送モードフィールドを形成する。

これら3つのビットは宛て先アドレス指定のためにどの
 フォームの転送を使用すべきかを表示する。ソース指定
 モードルックアップテーブルおよび値によるフィル操作
 は宛て先アドレス指定のために定義されていない（保留*
 *されている）。
 【0311】
 【表16】

ビット			宛て先転送モード
6	5	4	
0	0	0	次元の定められた転送
0	0	1	予約
0	1	0	予約
0	1	1	予約
1	0	0	可変パッチデルタガイド転送
1	0	1	可変パッチオフセットガイド転送
1	1	0	固定パッチデルタガイド転送
1	1	1	固定パッチオフセットガイド転送

表16

【0312】パケット転送オプションフィールドのビット
 ※行する。
 ト9～8は、ソース更新モードフィールドを形成する。
 【0313】
 表17に、このソース更新モードが示されている。適当
 【表17】
 なソース反転アドレス指定ビット、反転ソースCまたは
 反転ソースBがセットされる場合、加算よりも減算を実※

ビット		ソース更新オペレーション
9	8	
0	0	更新せず
0	1	ソース転送内の最終ラインの開始アドレスにBピッチを加算（これより減算）し、結果をソースに書き込む。
1	0	ソース転送内の最終パッチの開始アドレスにCピッチを加算（これより減算）し、結果をソースに書き込む。
1	1	ソース転送内の最終パッチの開始アドレスにCピッチを加算（これより減算）し、その結果をPTパラメータテーブル内のソースおよび開始アドレスに書き込み、次にPTオプションフィールド内の反転ソースCアドレス指定ビットをトゲリングする。

表17

【0314】これら2つのビットはパケット転送が完了した際に元のパケット転送パラメータ内のソース開始アドレスをどの値で更新すべきかを表示する。これらビットが0でなければパケット転送完了後に追加的ソースアドレス計算を実行する。次にパケット転送パラメータに指定された元のソース開始アドレス上にこの値を書き込む。これによりパケット転送を再び送ることができ、先の時間に残っていた場所より続けることができる。このような転送は、2つのメモリエリア間でピンポン操作するのに特に有効である。これらモードは先に宛て先アド

レスの発生で述べたように、操作アドレスの発生でも同じような用途がある。

【0315】パケット転送オプションフィールドのビット14～12は、ソース転送モードフィールドを形成する。これら3つのビットは、ソースアドレス指定のためにどのフォームの転送を使用すべきかを表示する。表18にこれらのコーディングを示す。

【0316】

【表18】

ビット			ソース転送モード
14	13	12	
0	0	0	次元の定められた転送
0	0	1	値でフィルする転送
0	1	0	予約
0	1	1	固定パッチオフセットガイドLUT
1	0	0	可変パッチデルタガイド転送
1	0	1	可変パッチオフセットガイド転送
1	1	0	固定パッチデルタガイド転送
1	1	1	固定パッチオフセットガイド転送

表18

【0317】バケット転送オプションフィールドのビット18～16は、バケット転送アクセスモードフィールドを形成する。これら3つのビットは表19に示されるような特別なアクセスモードをエンコードするのに用いられる。これらモードは宛て先にソースデータを書き込*

*む方法を変えるものであり、オンチップメモリの宛て先に対し、モード000の通常の転送しか許可されない。

【0318】

【表19】

ビット			バケット転送アクセスモード
18	17	16	
0	0	0	通常ページモード
0	0	1	周辺デバイス転送
0	1	0	ブロック書き込み
0	1	1	シリアルレジスタ転送
1	0	0	8ビットソースストランスペアレンシー
1	0	1	16ビットソースストランスペアレンシー
1	1	0	32ビットソースストランスペアレンシー
1	1	1	64ビットソースストランスペアレンシー

表19

【0319】000のコーディングは、外部メモリに対する通常のアクセスモードまたはページモードアクセスを可能にする。ソースまたは宛て先上では特別なアドレス指定モードは使用されない。変更をすることなく、ソースから宛て先にデータが転送される。

【0320】001のコーディングは、周辺デバイス転送モードを可能にする。周辺デバイスモードはメモリコントローラとして転送コントローラ80を使用するマルチプロセッサ集積回路100の外部のメモリを、他のデバイスが読み出ししたり書き込んだりするのを可能にする。デバイスのメモリの読み出しはソースをプログラムすることによって達成され、デバイスのメモリへの書き込みは宛て先をプログラムすることによって達成される。いずれの場合にせよ、転送コントローラ80は通常は転送パラメータに従って発生されたアドレスによりメモリアドレスおよび制御ラインをドライブするが、周辺デバイスがデータを読み出ししたりドライブしたりできるように、データバスをハイインピーダンスモードとする。周辺デバイスモードは値でフィルする転送を除き、任意の形態のソースまたは宛て先転送と共に使用できる。

【0321】010のコーディングはバケット転送がVRAMブロック書き込みを使用できるようにする。これにより、転送コントローラ80はVRAMカラーレジスタにロードし、VRAMブロック書き込みモードを使って外部メモリへの宛て先書き込みを実行する。このモー

ドでは、VRAMカラーレジスタデータがVRAM内のどの位置に書き込まれるかを指定する、ブロック書き込みアドレスマスクビットをソースデータが表示する。これらビットは通常のアドレス指定を用いてソースからフェッチされ、その後、ブロック書き込みモードを用いて宛て先VRAMに書き込まれる。VRAMカラーレジスタにロードされる値は、バケット転送パラメータのカラーレジスタ値として指定される。

【0322】ブロック書き込みオペレーションはオフチップの宛て先だけにサポートされている。オンチップ宛て先アドレスへのブロック書き込みの試みは、エラー条件によりバケット転送を一時中断させる。

【0323】011のコーディングはシリアルレジスタ転送モードを可能にする。一般に、VRAMのバルク初期化を行うために、シリアルレジスタ転送モードが使用される。VRAMの行をVRAMシリアルシフトレジスタにコピーするのに、ソースアドレスが使用される。次にシフトレジスタを多数のVRAMメモリの行にコピーするのに、宛て先アドレスが使用される。通常のオペレーションはソースへのカウントに1バイトをロードし、宛て先に1のAカウントおよびn-1（ここでnは書き込みを行う行の数）のBカウントをロードすることである。このモードでは、データバスまたはクロスバー50上でのデータ転送は行われず、VRAMによりすべてのデータ転送が行われることに留意されたい。すべてのソースアクセスおよび宛て先アクセスは非ページモードと

なる。シリアルレジスタ転送オペレーションは、オフチップのソースおよび宛て先だけにサポートされている。

【0324】1XXフォームのコーディングはトランスペアレンシーを可能にする。通常、ソースおよび宛て先の次元の定められた転送またはガイド転送が実行される。しかしながら宛て先データを書き込む前に、このデータはパケット転送パラメータ内に示されたトランスペアレンシー値と比較される。このトランスペアレンシーの比較は、トランスペアレンシーおよびバイト書き込み回路353で行われる。パケットアクセスモードの2つの最小位のビットはトランスペアレンシーデータのサイズを表示する。従って、1回の64ビットの比較、2回の32ビットの比較、4回の16ビットの比較または8回の8ビットの比較がなされる。比較のいずれかが真であれば、トランスペアレンシーおよびバイト書き込み回路353は、宛て先バイトが書き込まれないように対応するバイトストローブをディスエーブルする。オンチップソースまたは宛て先アドレスによるトランスペアレンシー*

	バイト アドレス		バイト アドレス
ソース開始アドレス	PT+08 <=>	PT+12	転送先開始アドレス
ソースA/Bカウント	PT+16 <=>	PT+20	転送先A/Bカウント
ソースCカウント	PT+24 <=>	PT+28	転送先Cカウント
ソースBピッチ	PT+32 <=>	PT+36	転送先Bピッチ
ソースCピッチ	PT+40 <=>	PT+44	転送先Cピッチ

表20

【0327】PTにおける次のエントリーアドレスデータおよびPT+04におけるパケット転送オプションフィールドは、これらの値がソースまたは宛て先に関連していないので、それらの先の位置に止まることに留意されたい。更にトランスペアレンシーデータまたはカラーレジスタデータを記憶するPT+48およびPT+52は、スワップされないことにも留意されたい。これにより、64ビットのトランスペアレンシーまたはカラーレ※

*シーの試みにより、パケット転送はエラー条件で一時中断される。

【0325】パケット転送オプションフィールドのビット19は、交換用ソースおよび宛て先パラメータビットである。ビット19のセットにより、ソースおよび宛て先パラメータをマニュアルでスワップすることなく、パケット転送方向を反転できる。これは元の位置にデータを戻す際に有効である。交換用ビットがセットされると、転送コントローラ80はパケット転送パラメータをロードする際に、すべてのソースおよび宛て先値の開始アドレス、ピッチ、カウント、ガイドテーブルポインタおよびフィル値のすべてをスワップする。表20は、このビットがセットされる場合の32ビットのスワップを示す。PTはパケット転送パラメータテーブルの次のエントリーアドレスのアドレスを表示する。

【0326】

【表20】

※ジスタパラメータはその値を維持できる。転送コントローラ80はソースおよび宛て先に関連したパラメータワードのスワップのほかに、パケット転送オプションフィールド内でソースおよび宛て先に関連したビットもスワップする。これについては表21に示されている。

【0328】

【表21】

	ビット数		ビット数
転送先更新モード	0 <=> 8		
	1 <=> 9	ソース更新モード	
転送先転送モード	4 <=> 12		
	5 <=> 13	ソース転送モード	
転送先反転Bアドレス指定	6 <=> 14		
	24 <=> 21	ソース反転Bアドレス指定	
転送先反転Cアドレス指定	25 <=> 22	ソース反転Cアドレス指定	

表21

【0329】スワップのいずれかの結果、機能がサポートされなくなると、パケット転送はエラー条件により一時中断する。パケット転送パラメータがロードされるときはいつも、ソースパラメータと宛て先パラメータとの交換が実行される。パケット転送が一時中断されている場合、そのときのパラメータはリクエスト中のプロセッサのパラメータメモリにセーブされる前に、元の位置へスワップし戻される。一時中断されたパケット転送がレ

ストアされる場合、パラメータは転送コントローラ80によりロードされる際に再びスワップされる。

【0330】パケット転送オプションフィールド内で更新モードのうちの一つが指定される場合、パケット転送の完了時に通常のオペレーションが行われる。例えばソース更新オペレーションが選択される場合、元のパケット転送パラメータ内ソース開始アドレスは、パケット転送中に宛て先開始アドレスとして実際に使用されても更

新されることになる。同様に、更新モードとしてトグル反転ソースCアドレス指定ビットを指定すると、元のパケット転送オプションフィールドのビット22がトグルリングされ、これにより実際にパケット転送が再送信された場合、宛て先Cのアドレス指定が反転される。

【0331】上記例が示すようにXビットを使用するには注意が必要である。例えば次元の定められた宛て先パケット転送に対する値で満たされたソースのために指定する場合、値によるフィルオペレーションは宛て先オペレーションとして指定できないので、パケット転送はエラーにより一時中断される。エラーが発生しない場合でも、宛て先開始アドレスおよびカウントには、一般にプログラムされないソース値がロードされる。同様に、フィル値のワードには宛て先ピッチがロードされる。この結果、極めて無意味なパケット転送となる。

【0332】パケット転送オプションのビット21は反転ソースBアドレス指定ビットである。このビットを1にセットすると、ソースの第2の次元が後方にアドレス指定される。加算よりもむしろ先のライン開始アドレスからBピッチが減算される。このビットが0であれば、通常はパケット転送ソースアドレス指定が行われる。

【0333】パケット転送オプションフィールドのビット22は、反転ソースCアドレス指定ビットである。このビットを1にセットすると、ソースの第3次元が後方にアドレス指定される。加算よりもむしろ先のパッチ開始アドレスからCピッチが減算される。このビットが0であれば、通常はパケット転送ソースアドレス指定が行われる。このビットはガイド転送に対しては意味がなく、ガイドテーブルを介してCのアドレス指定を特定する。

【0334】パケットテーブルオプションフィールドのビット23は、反転ソースAアドレス指定ビットである。このビットを1にセットすると、ソースおよび宛て先の第1次元は後方にアドレス指定される。ダブルワードアドレスはインクリメントされる代わりにデクリメントされる。このことは、Bピッチが加算（または減算）される値は、第1の次元における最も大きなアドレスであることを意味している。ダブルワード内でのバイトによるアドレス指定は反転されず単なるダブルワードのアドレス指定であることに留意されたい。

【0335】パケットテーブルオプションフィールドのビット24は、反転宛て先Bの宛て先指定ビットである。このビットを1にセットすると、宛て先の第2の次元が後方にアドレス指定される。Bピッチは加算されるよりもむしろ先のライン開始アドレスから減算される。このビットが0であれば、通常はパケット転送ソースアドレス指定が行われる。

【0336】ビット25は反転宛て先Cのアドレス指定

ビットである。このビットを1にセットすると、宛て先の第3の次元が後方にアドレス指定される。Cピッチは加算されるよりもむしろ先のパッチ開始アドレスから減算される。このビットが0であれば、通常はパケット転送ソースアドレス指定が行われる。このビットはガイド転送に対しては意味がなく、ガイドテーブルを介してCアドレス指定を特定する。

【0337】パケットテーブルオプションフィールドのビット28は、ビットを終了した際のインタラプトである。このビットを1にセットすると、リンクされたリスト上のこのエントリを終了した後にすぐに、パケット転送を開始しているプロセッサに通常のインタラプトが送られる。リンクされたリストは更にエントリを含むことができる。これによりリンクされたリスト内の特定のポイントに達したときに、リクエスト中のプロセッサにフラグを立てることができる。このビットが0であれば、リンクされたリスト内の対応するパケットエントリが終了した際には、プロセッサにはインタラプトは送られない。従って、このビットセットを有するエントリが生じて完了するか、またはリンクされたリストの終わりにあるパケット転送が完了するかのいずれかまでに、インタラプトが発生されることはない。しかしながらエラーがある時間に生じた場合、転送コントローラ80はリクエスト中のプロセッサに即座にエラーインタラプトを送る。

【0338】パケット転送オプションフィールドのビット30～29は、パケット転送ステータスフィールドを形成する。パケット転送ステータスフィールドはパケット転送リクエストのステートを表している。プロセッサがリクエストを送る際には、これらビットは常に00とセットされていなければならない。リンクされたリスト内のパケット転送が一時中断状態の場合、転送コントローラ80は、これが中断中のパケットパラメータエリアにセーブしているパケット転送オプションフィールド内の適当なパケット転送ステータスビットをセットする。中断されたパケット転送がフレッシュなパケット転送よりも多くのパケットを含んでおり、従って異なる態様でセーブされ、レストアされるので、このような操作が必要である。転送コントローラ80がパケット転送パラメータをロード中に、これらビットのいずれかが1とみなされると、転送コントローラはパケット転送を一時中断し、よって余分なパラメータのすべてをレストアする。このフィールドは、表22にリストされているようにコード化される。

【0339】

【表22】

ビット		バケット転送ステータス
30	29	
0	0	一時中断されていない
0	1	一時中断であるがフォールトでない
1	0	一時中断、ソース上でフォールト
1	1	一時中断、宛て先上でフォールト

表22

【0340】次の条件のうちの1つが生じたためにバケット転送が一時中断されると、転送コントローラ80はこれらビットに01を書き込む。すなわち転送コントローラ80により、より高い優先度のバケット転送リクエストが受信される場合、バケット転送が多くタイムアウトしてしまった場合、バケット転送をリクエストしたプロセッサがバケット転送の一時中断を求めた場合、またはエラー条件が生じた場合、ビット30は実際にフォールトが生じたことによりバケット転送を一時中断したことを表示する。ビット29は、フォールトがソースにあるのか宛て先にあるのかを表示する。フォールト条件を解決するため、この情報はマスタプロセッサ60が必要とする。このことは、フォールトの生じたバケット転送のパラメータを再ロードする際の転送コントローラ80には関係がない。その理由は、一時中断プロセスとレストアッププロセスとはすべてのタイプの一時中断バケット転送に対して同じであるからである。

【0341】交換用ソースおよび宛て先パラメータビット(ビット19)が中断されたバケット転送オプションフィールド内で1であれば、フォールトの生じたバケット転送に対するビット29の意味は反転する。従って10が宛て先上でのフォールトを示し、11がソース上でのフォールトを示す。従ってマスタプロセッサ60のソフトウェアはフォールトの生じたアドレスを決定する際に、ビット29と19の双方を検査しなければならない。

【0342】転送コントローラ80はバケット転送中にフォールトが生じた場合に数種の措置をとる。転送コントローラ80は、ビット29と30を適当にセットし、この適当なビットをFLTSSTSレジスタ内にセットする。転送コントローラ80がマスタプロセッサ60にフォールトインタラプトを発生する。デジタル画像/グラフィックプロセッサ71、72、73または74のうちの1つがバケット転送を発生する場合、このプロセッサはフォールトが生じていることについて認識していない。マスタプロセッサ60は、このフォールト条件をクリアし、FLTSSTSレジスタ内のフォールトフラグをクリアしなければならない。FLTSSTSレジスタ内の対応するフォールトフラグをクリアする際に転送コントローラ80はバケット転送を自動的に再送信する。

【0343】バケット転送オプションフィールドのビット31はストップビットである。このビットはリンクされたリストの終了部をマークするのに使用されている。このビットが1であるバケット転送に会うと、バケット

転送は完了され、リンクされたリストが終了される。終了前にリクエスト中のプロセッサのパラメータメモリ内のリンクされたリストの開始アドレス位置に、次のエントリーアドレスフィールドがコピーされる。従ってリンクされたリストが再イネーブル化されると、リンクされたリスト内の次のエントリーで実行が開始する。これにより、ピンポンオペレーションのような繰り返しオペレーションに特に有効な円形状のリンクされたリストが形成できる。ストップビットは所望の位置でリンクされたリストをブレイクするのににも使用できる。

【0344】転送コントローラ80がパラメータテーブルからのバケット転送パラメータを一旦読み出すと、データの転送を開始する準備が完了する。これを行うため、転送コントローラ80はクロスバアクセスまたは外部メモリアccessのいずれかまたは双方を発生しなければならない。ソースから宛て先へのデータの基本的フローには4つの可能性がある。すなわちオンチップからオンチップへ、オンチップからオフチップへ、オフチップからオンチップへ、更にオフチップからオフチップへのフローがある。最初の3つは通常取り扱われているが、最後のケースは特別なケースである。

【0345】ソースマシン320は、正常なバケットデータ転送フロー中に、転送パラメータに基づきソースアドレスを発生し、これを用いて適当なオンチップまたはオフチップメモリからデータをフェッチする。クロスバまたは外部メモリバスからデータが受け取られると、必要なバイトは抽出され、ソース整列ロジック332によって整列され、次にバケット適当なFIFOバッファ312に入れられる。これと同時に宛て先マシン340は宛て先メモリのためのアドレスも発生する。バケット転送FIFOバッファ311が一旦次の宛て先メモリアccessに必要なバイト数を含むと、宛て先マシン340は必要なクロスバまたは外部メモリスイクルを発生する。

【0346】バケット転送FIFOバッファ311は、データフローを制御し、ソースマシン320と宛て先マシン340との同期を維持するように働く。あるときに、バケット転送FIFOバッファ311が次の宛て先accessのための十分なソースバイトを含んでいない場合、データが利用可能となるまで宛て先マシン340は停止する。これと同じように、バケット転送FIFOバッファ311がフル状態となれば、宛て先マシン340が次のソースaccessを完了させるのに十分なバイトを引き出すまで停止する。これにより、ソース転送が宛て

先転送をオーバーランさせることが防止される。特殊なパケット転送アクセスモードに対しては、このデータフローは多少変えることができる。

【0347】クロスバー50と外部メモリインターフェースは独立しているため、ソース転送は外部バス上で行われ、クロスバー上で宛て先転送を行うことができ、また、パラレル状態で外部バス上で宛て先転送を行い、クロスバー上でソース転送を行うこともできる。オンチップからオンチップへの転送に対してはソースと宛て先とは必要に応じてクロスバーインターフェースを共用し、サイクルをインターリーブすることになる。

【0348】キャッシュサービスリクエスト、ダイレクト外部アクセスリクエスト、フレームコントローラ90のリクエスト、緊急リフレッシュおよびホストリクエストは、パケット転送を一時中断させることはない。これらリクエストがパケット転送よりも優先度が高い場合のみ、ソースマシン320および宛て先マシン340の一方または双方を停止できる。フレームコントローラ90および緊急リフレッシュサイクルは、外部メモリインターフェースだけを使用する。従ってパケット転送クロスバーアクセスは続いて行うことができる。ソースマシン320または宛て先マシン340のいずれかが、外部メモリインターフェースを使用している場合、パケット転送FIFOバッファ311は、最終的にフル状態または空状態となる。従ってクロスバーを用いるコントローラは外部メモリインターフェースが再び利用できるまで停止していなければならない。オンチップからオンチップへの転送である場合、ソースマシン320および宛て先マシン340の双方は、障害を受けない状態を続けることができる。キャッシュおよびダイレクト外部アクセスサービスは、クロスバーと外部インターフェースの双方を使用するので、これらは一般に、必要とするサイクル数の間でパケット転送を停止する。

【0349】オンチップソースからオフチップの宛て先へのパケット転送は、他の3つのパケット転送の例と異なって取り扱われる。DRAMまたはVRAM上でページモードの利点を活用するため、転送コントローラ80はオフチップソースからオンチップへの列アクセスのページモードバーストを実行し、次にオンチップからオフチップの宛て先への別のページモードバーストを実行する。これを行うにはオンチップバッファを使用しなければならない。各プロセッサは図15に示すように、この目的のために保留された対応するパラメータメモリ内の128バイトのエリアを有する。図15に示すように、マスタプロセッサ60によってリクエストされるオフチップからオフチップへのパケット転送は、アドレスの16進数01010100から16進数0101017Fを使用する。デジタル画像/グラフィックプロセッサ71、72、73、74のうちの一つによってリクエストされるオフチップからオフチップへのパケット転送が、

アドレスの16進数0100#100から16進数0100#17F（ここで#は表3にリストされているようなデジタル画像/グラフィックプロセッサの番号に対応する）を利用する。このパラメータメモリバッファの内外へのデータの転送は、転送コントローラ80のハードウェアにより処理され、ユーザーに対しトランスペアレントである。

【0350】パケット転送パラメータの組み合わせは、完了するのに極めて長い時間を必要とする、極めて大きな転送の指定を可能とする。更により高い優先度のパケット転送リクエストが、パケット転送をインタラプトし続け、パラメータのロードに長い時間がかかるの防止する。このような状況のいずれかの発生を防止するため、転送コントローラ80は図31に示されている2つの24ビットレジスタのパケット転送最小レジスタPTMIN511とパケット転送最大レジスタPTMAX512を含むタイマーを有する。これらレジスタはパケット転送の最小長さと最大長さを指定する。パケット転送カウンタPTCOUNT513は、パケット転送を実行するクロックサイクル数を表示する。

【0351】パケット転送最小レジスタPTMIN511は、より高い優先度のパケット転送リクエストによって中断できる前に、パケット転送が実行しなければならないクロックサイクルの最小数を示す。パラメータがロードされた後にパケット転送が開始すると、タイマーシーケンサ515はパケット転送最小レジスタPTMIN511に記憶されていた値を、パケット転送カウンタPTCOUNT513にロードする。パケット転送カウンタPTCOUNT513は、パケット転送が転送コントローラ80によってアクティブにサービスされるクロックサイクルごとに、1だけデクリメントする。パケット転送カウンタPTCOUNT513は、パケット転送中に生じ得るキャッシュサービスサイクル、フレームコントローラ90サイクル、ホストサイクルまたはリフレッシュサイクル中にはデクリメントしないが、アクティブなパケット転送サービス中に生じる再試行または待機ステート中にデクリメントする。しかしながら、クロスバーのパケット転送アクセスがまだ行われている場合、外部バス上でのフレームコントローラ90、ホストまたはリフレッシュ活動中にデクリメントされる。パケット転送はPTCOUNTが0に達したことを0検出器514がタイマーシーケンサ515に信号を送るまで、優先度の高いパケット転送リクエストによってインタラプトすることはできない。またパケット転送はエラーまたはフォールト条件によって一時中断することもできる。パケット転送最小レジスタPTMIN511には、リセット時に16進数10000（65,536サイクル）がロードされる。

【0352】パケット転送最小レジスタPTMIN511の重要な用途は、すでに中断したパケット転送のパラ

メータを別のメモリエリアに転送するための停止不能なパケット転送を行うことである。別の一時中断によりデータを転送するパラメータメモリエリアにオーバーライトされるので、パケット転送最小レジスタPTMIN511はパケット転送を完了できるように保証するために使用される。かかる転送はフォールト状態にならないことが重要である。その理由は、この状態になるとパラメータメモリエリアもオーバーライトされるからである。このような特徴により、最小時間前により高い優先度のメモリアクセスがパケット転送をアボートすることが防止され、従って最小の数のデータ転送が行われる。

【0353】パケット転送最大レジスタPTMAX512は、1つのパケット転送によるデータの転送の独占を防止するのに使用される。PTMIN時間が経過し、パケット転送カウンタPTCOUNT513が0検出器514に検出されるように、0にデクリメントされると、タイマーシーケンサ515はパケット転送最大レジスタPTMAX512内に記憶されていた値をパケット転送カウンタPTCOUNT513にロードする。これにより、タイムアウトするまでに転送を進めることができる残りの時間が決まる。従ってインタラプトされないパケット転送のための最大期間は、PTMIN+PTMAXクロックサイクルとなる。パケット転送カウンタPTCOUNT513は、リフレッシュのような非パケット転送サイクルを除く、パケット転送が連続的にアクティブとなるサイクルごとにデクリメントされる。パケット転送完了前にパケット転送カウンタPTCOUNT513内の値が0に達したことを、0検出器514が検出すると、パケット転送はタイムアウトしたものと見なされる。タイマーシーケンサ515は、パケット転送を一時中断し、転送コントローラ80はラウンドロビン状に同じ優先度の次のリクエストに移る。同じ優先度の他のリクエストがペンディング中となっていなければ、このような同じ優先度のリクエストまたは優先度のより高いリクエストが生じるまで、または転送が完了するまで、0のPTCOUNTと共に転送を続行できる。パケット転送最小レジスタPTMIN511内に指定されたサイクル数が経過した後に、優先度のより高いパケット転送リクエストが受信されると、PTMAXに達したか否かに係わらず、アクティブなパケット転送が一時中断される。パケット転送最大レジスタPTMAX512にはリセットで16進数10000(65,536サイクル)がロードされる。

【0354】パケット転送が一時中断状態となると、リンクされたリスト全体も一時中断される。ラウンドロビントークンはリンクされたリスト内の次のパケット転送でなくて、ペンディング中のリクエストと共に次のプロセッサに進む。パケット転送がタイムアウトすると、一時中断された転送のステートはリクエスト中のプロセッサのパラメータメモリにセーブされる。ラウンドロビン

の優先度がそのプロセッサに戻されると、転送コントローラ80は連続のためのこのリクエストを自動的に再送信する。一時中断されたパケット転送が再開されるときはいつも、フル状態のPTMINおよびPTMAX値が有効となる。

【0355】好ましい実施例では、パケット転送最小レジスタPTMIN511およびパケット転送最大レジスタPTMAX512は、24ビットしか含まない。従ってタイムアウトすることなく、パケット転送サービスが続くことのできる最大時間は50MHzの目標作動周波数で約0.67秒である。

【0356】パケット転送用のリンクされたリストは、次のように転送コントローラ80によって管理される。緊急優先度を除くパケット転送中にリフレッシュ、フレームコントローラ90、ホストインターフェースまたはキャッシュサービスリクエストが受信される場合、パケット転送パラメータのステートは転送コントローラ80の内部レジスタに保持され、必要であればソースおよび/または宛て先転送が停止される。優先度のより高いリクエストのサービスが完了すると、パケット転送が開始される。

【0357】優先度のより高いパケット転送リクエスト、タイムアウト、リクエスト中のプロセッサからの一時中断リクエスト、フォールトまたはエラーによってパケット転送がインタラプトされると、パケット転送が一時中断される。リクエスト中のプロセッサのパラメータメモリ内のリンクされたリストの開始アドレスが、セーブされたパケット転送パラメータをポイントするように変更される。このインタラプトが優先度のより高いパケット転送によるものである場合、ラウンドロビントークンはインタラプトされたパケット転送と共に留まるので、優先度のより低いリクエストが再開されると、そのサービスが再開される。パケット転送パラメータのローディング中に優先度のより高いリクエストが生じると、ローディングが停止される。一時中断は行われぬ。優先度のより高いパケット転送が完了すると、元のパラメータリストからパケット転送パラメータがロードされる。パケット転送がタイムアウト、フォールト、エラーまたは一時中断リクエストにより一時中断されている場合、優先度決定チェーンの終了部にインタラプトされたパケット転送を送るように、ラウンドロビントークンが進められる。

【0358】パケット転送が完了し、パケット転送オプションフィールドのインタラプトビットが1となると、転送コントローラ80はリクエスト中のプロセッサにパケットの終了部のインタラプト信号を発生する。リンクされたリスト内に1のパケット転送オプションフィールドの停止ビットを有する最終パケット転送が完了すると、転送コントローラ80がリクエスト中のプロセッサにパケットの終了部インタラプト信号を発生する。パケ

ット転送が完了すると、パケット転送のパケット転送オプションフィールドストップビットがセットされているも、リクエスト中のプロセッサのパラメータメモリ内のリンクされたリストの開始アドレス位置に、パケット転送パラメータからの次のアドレスフィールドが書き込まれる。

【0359】パケット転送パラメータがロードされると、転送コントローラ 80 はパケット転送オプションフィールドのパケット転送ステータスビットをチェックする。パケット転送が一時中断されたことをこれらビットが表示する場合、一時中断された転送を付加的ステート情報がロードされる。これらについては後に更に説明する。

【0360】パケット転送がエラー状態を経験すると、転送コントローラ 80 は即座にデータの転送を停止し、一時中断を実行する。転送コントローラ 80 はリクエスト中のプロセッサの PTEROR フラグをセットする。転送コントローラ 80 はリンクされたリストが終了したことを表示するため、リクエスト中のプロセッサの COMN レジスタ 120 の Q ビットもセットする。リクエスト中のプロセッサは、セーブされたパラメータからのエラーの原因を決定するようにプログラムすることができる。

【0361】パケット転送中に多数の条件のいずれかがエラーを生じさせる。試みられた VRAM アクセスモード、例えばオンチップメモリ内のソースまたは宛て先によるブロック書き込み宛て先転送またはシリアルレジスタ転送がエラーを生じさせる。トランスベアレンシーを利用して試みられるオンチップ宛て先アクセスもエラーを生じさせる。値でフィルされた場合を除くソース転送、シフトレジスタ転送または周辺デバイス転送の長さよりも、パケット転送リクエスト宛て先転送のバイトの総数の長さが長くなると、エラーが生じる。ルックアップテーブルの宛て先オペレーションを試みる際にエラーが生じる。このようなエラーは、交換用ソースおよび宛て先パラメータのオペレーションから生じる。リンクされたリストの開始アドレスまたはリンクされたリストの次のエントリーがアドレスのオフチップをポイントする際にエラーが生じる。パケット転送パラメータテーブルおよびリンクされたリストの開始アドレスが 64 バイトの境界に整列していないときにも、エラーが生じる。ソースまたは宛て先ガイドテーブルポインタがオフチップメモリをポイントしたり、これらが正しく整列されていないときに試みられるパケットリクエストの際にエラーが生じる。実際のエラー状態を表示するようにステータスビットはセーブされず、このエラー状態は一時中断パラメータのステートから推定しなければならない。

【0362】リンクされたリストの開始アドレスまたはリンクされたリストのアドレスのオフチップの次のエントリー、整列状態のパケット転送パラメータテーブルま

たはオフチップメモリへのガイドテーブルポインタによって生じたエラーは、パケット転送パラメータをリクエスト中のプロセッサのパラメータメモリの一時中断エリアにセーブしない。これはデータを転送する前にパケット転送がアボートし、実際に開始することがないからである。従って、エラーの原因を診断しようとする、リンクされたリストのアドレスの有効性をチェックして、一時中断されたパラメータが有効であることを保証しなければならない。

【0363】転送コントローラ 80 は多くの条件下でパケット転送を一時中断する。優先度のより高いパケット転送リクエストを受けると、現在のパケット転送のために PTMIN に指定されていたサイクル数がなくなる場合に、パケット転送を一時中断する。転送コントローラ 80 はサイクル数が PTMIN+PTMAX を越え、タイムアウトし、別のプロセッサからの同じ優先度のパケット転送リクエストがペンディング中である場合、パケット転送を一時中断する。転送コントローラ 80 がリクエスト中のプロセッサがこのコントローラに COMM レジスタ 120 の S ビットにより一時中断することを求める場合、パケット転送を一時中断する。ソースアドレス指定または宛て先アドレス指定中にメモリフォールトが生じる場合、転送コントローラ 80 はパケット転送を一時中断する。エラー状態が検出される場合、転送コントローラ 80 はパケット転送を一時中断する。また、外部メモリアクセス中に再試行が行われ、別のプロセッサからの同じ優先度のパケット転送リクエストが待機中であり、現在のパケット転送のための PTMIN 内に指定されたサイクル数がなくなる場合、転送コントローラ 80 はパケット転送を一時中断する。

【0364】これらケースの各々における一時中断機構は同一である。転送コントローラ 80 は現在のパケット転送パラメータおよび転送コントローラ 80 の内部ステートをセーブする。この情報により、パケット転送を将来続行できる。これらパラメータはリクエスト中のプロセッサのパラメータメモリの一時中断エリアにセーブされる。図 15 に示すように、マスタプロセッサ 60 がパケット転送をリクエストした場合、このエリアは 16 進数 01010000 から 16 進数 0101007F となる。デジタル画像/グラフィックプロセッサ 71、72、73、74 の一つにリクエストされたパケット転送のための一時中断されたパケットパラメータエリアは、アドレスの 16 進数 0101#0000 から 16 進数 0100#07F (ここで # は表 3 にリストしたようなデジタル画像/グラフィックプロセッサ番号に対応する) を使用する。図 32 および 33 には、これらパラメータのためのフォーマットが示されている。図 32 における % のマークのついたダブるワードは、現在のエンディアンのためには調節されないことに留意されたい。

【0365】一時中断条件が生じ、ペンディング中の外

部メモリ行アクセスが完了した直後に、一時中断が開始する。リクエスト中のプロセッサのパラメータメモリ上の外部間バッファは空ではないが、パケット転送FIFOバッファ311の現在ステートがセーブされる。この方法により、原因にかかわらず、パケット転送は一貫して中断できる。パケット転送FIFOバッファ311の空状態は生じないので、この一時中断は高速となるようにも保証されている。従ってパケット転送FIFOバッファ311を空にするのに潜在的に低速のパケット転送を待つことなく、緊急優先度のパケット転送リクエストを急速にサービスできる。

【0366】パラメータメモリが一時中断の原因となっている場合、より高い優先度の転送コントローラ80のクロスバーの優先レベルで実行される。従って一時中断が完了するまでデジタル画像/グラフィックプロセッサ71、72、73、74またはマスタプロセッサ60は一時的な競合を経験することがある。一時中断がタイムアウト、フォールト、エラーまたはリクエスト中のプロセッサからの一時中断リクエストによるものであるときは、パケット転送の元の転送コントローラ80のクロスバーの優先度で、この一時中断が実行される。

【0367】新しいアドレスをポイントするように、リンクされたリストの開始アドレスポイントが変えられることを条件に、一時中断されたパケット転送パラメータを他の場所でコピーし、その新しい位置から再送信することができる。パケット転送オプションフィールドのパケット転送ステータスビットは、パケット転送が一時中断されたことを表示するので、一時中断されたパラメータの全組は、その位置がどこであれ、再送信時にロードされる。一時中断されたパラメータの新しい開始アドレスは、偶数の128バイトの境界上で整列していなければならない、従って7つの最小位アドレスビットは、再送信すべき一時中断されているパケット転送に対し00000000とならなければならない。

【0368】パケット転送一時中断エリアにセーブされた最初の8つの64ビットダブルワードは、通常のパケット転送パラメータを示す。これらは、少数の例外を除き、ユーザーによってプログラムされたパラメータと同じである。最初の32ビットワードは、リンクされたリスト上の次のエントリーのアドレスでなく、一時中断されたパケット転送の元のエントリーアドレスを含む。Cカウンタフィールドは、次元の定められた転送のための現在のCカウンタを含む。ガイド転送に対し、Cカウンタフィールドはガイドカウンタとガイドテーブルポインタフィールドを含み、ガイドテーブルポインタフィールドはガイドテーブルポインタの現在位置を含む。更にパケット転送オプションフィールドはパラメータが一時中断されたパケット転送を表示することを示すように変更された、そのパケット転送ステータスフィールドを有する。

【0369】一時中断エリアは通常のユーザーがプログラムしたパケット転送パラメータの他に8個の保留された内部ステートのダブルワードも含む。これらワードは一時中断されたパケット転送が再スタートされる際に自動的にロードされるフィールドである。これらは一時中断した際のパケット転送の次元内ステートに関する情報を含む。これらの値は、これらが再送信される際に残される場所で正確に一時中断されたパケット転送を正しく開始できるようにセーブされる。16進数040で終了するアドレスにおけるデータワードは、現在のソースAおよびBのカウントを記憶する。これらはソースマシン320のACURRENTおよびBCURRENTからの16ビットの値である。Bのカウントはビット31～16に記憶され、Aのカウントはビット15～0に記憶される。16進数044で終了するアドレスにおけるデータワードは、現在の宛て先AおよびBのカウントを記憶する。これらは宛て先マシン340のAカウンタおよびBカウンタレジスタからの16ビットの値である。Bのカウントはビット31～16に記憶され、Aのカウントはビット15～0に記憶される。16進数048で終了するアドレスにおけるデータワードは、現在のソースA開始アドレスまたは現在のガイドテーブルアドレスである。一時中断されたパケットサービスパラメータ内の対応するPビットは、どのデータがここに記憶されたかを表示する。元のBのカウントが0に達するまでガイドテーブルのフェッチは行われないので、ガイドパケット転送でもここにソースAの開始アドレスを記憶できることに留意されたい。16進数04Cで終了するアドレスにおけるデータワードは、現在の宛て先Aの開始アドレスまたは現在のガイドテーブルフェッチ値であり、その識別は一時中断されたパケットサービスパラメータの対応するPビットによって決定される。16進数050および056で終了するアドレスにおけるデータワードは、現在の宛て先Bの開始アドレスまたは対応するガイドテーブルフェッチ値である。

【0370】16進数058で終了するアドレスで開始する16ビットのダブルワードは、一時中断されたパケットサービスパラメータを記憶する。一時中断されたパケットサービスパラメータのフォーマットは図32および33に示されている。上部データワードのビット31～30はPビットである。ビット31における1は、ガイドテーブルからフェッチされたソースマシン320のASTARTレジスタおよびBSTARTレジスタの値を記憶する。これと異なり、これらレジスタは次元の定められた転送からの値を記憶する。同様にビット30における1は、一時中断されたパケット転送パラメータがガイドテーブルからフェッチされたソースマシン320のASTARTレジスタおよびBSTARTレジスタ値を記憶し、0はこれら値が次元の定められた転送からのものであることを表示する。これらビットは一時中断さ

れたパケット転送がガイドソースまたは宛て先アドレス指定を含んでいたことを、パケット転送オプションフィールドが表示している場合に意味があるにすぎない。これとは異なり、データは次元の定められた転送からのものであると見なされる。

【0371】Eビット（上部データワードのビット29）は、一時中断されたパケット転送がオフチップからオフチップへのパケット転送であることを表示する。かかるオフチップからオフチップへのパケット転送は、リクエスト中のプロセッサのパラメータメモリ内のバッファを利用する。このEビットが1であれば、一時中断されたパケット転送はオフチップからオフチップへのパケット転送である。このEビットが0であれば、一時中断されたパケット転送は他の3つのタイプのうちの一つであり、リクエスト中のプロセッサのパラメータメモリは外部から外部へのバッファ化されたデータを記憶しない。

【0372】Sビット（上部データワードのビット28）は、パラメータメモリの外部間バッファがソースマシン320によってアクセスされているか、宛て先マシン340でアクセスされているかどうかを表示する。Eビットが1であって、リクエスト中のプロセッサのパラメータメモリ内の外部間でバッファ化されたデータの記憶を表示している場合にかぎり意味がある。Sビットが1である場合、ソースマシン320はパケット転送が一時中断されているときに、外部間バッファにアクセス中である。Sビットが0であれば宛て先コントローラは外部間バッファにアクセス中である。

【0373】上部データワードのビット20～16は、BUFPTRレジスタ414の値を記憶する。BUFPTRレジスタ414は、パラメータメモリの外部間バッファへのポインタである。このポインタを記憶すると、パラメータメモリの外部間バッファのステータスをリカバーすることにより、外部間パケット転送の再使用が可能となる。当然ながらこのフィールドは、一時中断パケット転送が外部間パケット転送である場合のみ意味がある。

【0374】上部データワードのビット14～8は、BUF COUNTレジスタ441の値を記憶する。このBUF COUNTレジスタ441の値は、Sビットのステータスに基づいて解釈される。Sビットが1であって、パラメータメモリの外部間バッファのソースアドレス指定が一時中断していることが表示されると、BUF COUNTレジスタ441はバッファ内に残っているバイト数を表示する。Sビットが0であり、パラメータメモリの外部間アドレスの宛て先アクセスの一時中断を表示している場合、BUF COUNTレジスタ441はバッファ内に記憶されたバイト数を表示する。このフィールドは一時中断された外部間パケット転送にしか意味がない。上部データワードのビット3～0は、FCOUNT

である。これはパケット転送FIFOバッファ311にこのとき保持されているデータのバイト数である。

【0375】一時中断されたパケットサービスパラメータの下部データワードは、2つのFIFOポインタを保持する。下部データワードのビット11～8は、FDPTRすなわち宛て先ポインタを記憶する。下部データワードのビット3～0は、FSPTRすなわちソースポインタを記憶する。これらポインタは、一時中断点における一時中断パケット転送の再開を可能にする。

【0376】16進数060で終了するアドレスで開始する4つのデータワードは、パケット転送FIFOバッファ311の内容を記憶している。パケット転送FIFOバッファ311を空にするかわりに、このデータを記憶することにより、より高い優先度のパケット転送を早期にスタートできる。将来、より大きなバッファを使用する場合に、このような大きなパケット転送用FIFOバッファ311の内容の記憶をサポートするために、16進数070で終了するアドレスで開始する4つのデータワードが保留されていることに留意されたい。

【0377】転送コントローラ80は、多数の標準的でないパケット転送アクセスモードをサポートする。これらにはブロックライト、シリアルレジスタ転送、トランスペアレンシーおよび周辺デバイス転送が含まれる。これら非標準的パケット転送アクセスモードは、パケット転送パラメータのパケット転送オプションフィールドによりイネーブルされる。

【0378】パケット転送パラメータのパケットアクセスモードが010に等しいとき、このモードはブロック書き込みパケットアクセスモードを可能にする。転送コントローラ80は3つの異なるブロック書き込み機構、すなわち8x、3xおよびシミュレート機構をサポートする。システムハードウェアは、ブロックの書き込みが開始する時間にブロック書き込み機構を決定する。これによりソフトウェアはシステムが使用するブロック書き込みのタイプがどのタイプであるかとは無関係に、またはこれをアドレス指定されたメモリがサポートするか否かとは無関係に、ソフトウェアがブロック書き込みを利用できるようにする。しかしながらブロック書き込みモードの各々は、64ビットのバスサイズおよび8ビットの値（1～8）に対してしかサポートされていない。ブロック書き込みのための宛て先スタートアドレスはオフチップであり、64ビットに整理していなければならない。すなわち最小位の6つのアドレスビットは000000でなければならない。

【0379】転送コントローラ80によって使用されるブロック書き込みモードは、外部回路によりマルチプロセッサ集積回路100のBS[1:0]ピンに入力された値によって選択される。好ましい実施例では、ブロック書き込みは64ビットのデータバスに対してのみサポートされているので、これらバスサイズ入力ブロック

書き込みおよび負荷カラーレジスタサイクル中にブロック書き込み選択として使用される。表23は、BS [1:0]により選択されたブロック書き込みモードを示す。

【0380】

【表23】

BS [1:0]		ブロック書き込みモード
0	0	シミュレートされたモード
0	1	予約
1	0	4×
1	1	8×

表23

【0381】種々のVRAMサイズおよびアーキテクチャのために、多数の異なるブロック書き込み方法がある。以下の表記法は、種々のブロック書き込み方法；すなわちC×L×R（ここでCはカラーレジスタ当たりの列位置の数、Lはビットによるカラーレジスタの長さ、Rはカラーレジスタの数である）を示すのに使用されている。たいていの1MビットのVRAMは、256Kビット×7ビットデータアレイを有し、4×4×1のブロック書き込みをサポートしている。これらは1つの4ビットカラーレジスタを有し、各ブロック書き込みサイクルは、4メモリアレイ平面での4つの隣接する列位置へのカラーレジスタの書き込みを制御している。

【0382】8×（倍の）ブロック書き込みは、位置当たり8ビットに、アクセス当たり8つの行位置を（すなわち8×8×1または8×8×2）書き込みできるVRAMと共に使用できるようになっている。ソースデータの各ビットは、データバスD [63:0]上の1ビットに出力され、このサイクルでアクセスされる64個の列のうちの1つに対する8ビットVRAMカラーレジスタの書き込みをイネーブルしたりディスエーブルしたりする。各列は各VRAM内の8つのメモリアレイ平面の各々における列位置を表示している。従って1回のアクセスで64までのバイトのカラーレジスタデータを書き込みできる。

【0383】8×ブロック書き込みをサポートするビデオRAMは、ブロック書き込みサイクル中に最小位の3つの列のアドレスを無視する。従って、データバスは64ビット（8バイト幅）であるので、ブロック書き込みサイクルは常に64バイトの境界上で開始する。64バイトに整列されていない宛て先アドレスに対し、転送コントローラ80はソースビットを整列し、書き込みされていない64バイトのアクセス内の位置に対して喪失した0を発生する。次に転送コントローラ80は、ビットがデータバス上のVRAMの各々内の適当な列位置をアドレス指定するように、ビットを再マップ化する。

【0384】図34は、小エンディアンモードにおける

8×ブロック書き込みのためのデータ再マップ化プロセスを示す。最初の8つのソースビットは宛て先のうちの最初の8バイトを制御する。これらバイトは実際は、アクセスされている8個の8平面VRAMアレイの各々における最小位の列位置である。最小位バイトはブロック書き込み中にVRAMのD0入力（および16ビットVRAMのためのD8入力）によって制御されるので、ソースビット0～7はデータバスのそれぞれのビット0、8、16、24、32、40、48および56にマップ化され、これらは、アクセスされたVRAMのD0入力（16ビットデバイスのためのD0およびD8入力）を表示する。ソースビットの残りは同様にマップ化される。転送コントローラ80は、大エンディアン8×ブロック書き込みのための同様なマッピングを行う。大エンディアンモードではソース画像のうちの最小位ビットは左側の最大ビットであり、最高位ビットは右側の最大ビットである。マッピング機構は小エンディアンモードのために使用されているものと同じであるので、ソースビット0～7はそれぞれビット63、55、47、39、31、23、15および7にマップ化されている。これらビットはVRAMアレイ内でアクセスされている最小位列位置を表示する。これらバイトは、VRAMのD0入力（16ビットデバイスのためのD0およびD8入力）によって制御されているので、ブロック書き込みが正しく作動できるように、逆の順にVRAMに接続しなければならない。

【0385】データは逆の順に書き込みされ、読み出されるので、データバスを逆の順に接続しても通常の読み出しおよび書き込みに影響しない。VRAMのシリアルポートの内外にシフトされるビットは、逆の順となることをユーザーは認識しなければならない。このようにするには、適当なオペレーションを保証するため、出力デバイスまたは入力デバイスを接続する際にシリアルデータバスの順を反転しなければならないことがある。

【0386】4×ブロック書き込みモードは、アクセス当たり4列の位置、すなわち位置当たり4または8ビットのいずれかで、4×4×1、4×4×4、4×8×1、4×8×2を書き込みできるVRAMと共に使用するように設計されている。4×4のブロック書き込みの場合、ソースデータの各ビットは、データバスD [63:0]上の2ビットに出力され、このサイクルでアクセスされる64列のうちの2つに対する4ビットVRAMカラーレジスタのうちの2つの書き込みをイネーブルまたはディスエーブルする。各列はアレイのうちの4つの平面上の列位置を表示する。各カラーレジスタは4ビット幅しかないで、8ビットのピクセルを表示するのに2つのレジスタが必要である。従って各ソースビットは8ビットの書き込みが生じるように、隣接するニブルに書き込む2つのカラーレジスタを制御しなければならない。これにより1回のアクセスでカラーレジスタデー

タのうちの32バイトまでを書き込みできる。

【0387】4×ブロック書き込みをサポートするVRAMは、ブロック書き込みサイクル中に最小位の2つの列アドレスを無視する。従ってブロック書き込みは常に32バイト境界上で始まる。8×ブロック書き込みと同じように、転送コントローラ80は32バイトブロック内の宛て先開始アドレスのダブルワードによって指定されたデータにソースデータを整列し、書き込まれていないダブルワードに対し、喪失した0を満たす。

【0388】図35は、小エンディアンモードにおける4×ブロック書き込みのためのデータ再マップ化プロセスを示す。最初の8つのソースビットは宛て先の最初の8つのバイトを制御する。しかしながら4×4ブロック書き込みでは、各データ入力では4ビットしか書き込みしないので、完全なバイトを書き込みするように2つのデータバスピンに各ソースビットをマップ化しなければならない。従ってソースビット0～7はビット0と4、8と12、16と20、24と28、32と36、40と44、48と52ならびに56と60にそれぞれマップ化される。

【0389】4×8×1または4×8×2のブロック書き込みの場合、VRAMカラーレジスタは8ビット幅である。しかしながら書き込むべきバイトを選択するのに、VRAMの8個のデータ入力のうちの4つしか使用しない。従ってデータバスのうちの半分は使用せず、1回のアクセスで32バイトしか書き込みできない。しかしながら、D[63:0]上に実際に入力されるデータは同じであり、VRAMは他のどのニブルも無視するだけであることに留意されたい。大エンディアンモードでは、ソースビット0～7は外部データバスのうちのビット63と59、55と51、47と43、39と35、31と27、23と19、15と11ならびに7と3にマップ化される。8×大エンディアンブロック書き込みと同じように、正しいオペレーションを保証するようにVRAMにデータバスを逆の順序で接続しなければならない。

【0390】ブロック書き込みをサポートしないメモリデバイスに対し、転送コントローラ80はシミュレートされたブロック書き込みモードを提供する。このモードでは、データバス上にパケット転送パラメータ内に含まれていた64ビットのカラーレジスタ値が出力され、サイクル中にアドレス指定される8バイトのうちの1つをイネーブルまたはディスエーブルするよう、CAS'列アドレスストローブピンを制御することにより、各ソースデータビットはバイト選択信号として機能する。従ってブロック書き込みは実質的にはカラーレジスタ値がフィル値となるような、値でフィルするタイプの転送に変換される。次に、宛て先アクセスは、ソースデータによって指定されるように、あるバイトへの書き込みをディスエーブルする通常のページモードの64ビットの書き

込みサイクルとなる。

【0391】4×および8×ブロック書き込みサイクルを実行する前に、VRAMのカラーレジスタに正しい値をロードしなければならない。転送コントローラ80はパケット転送パラメータに含まれるカラーレジスタ値を使用するロードカラーレジスタLCRサイクルを実行することによりこれを行う。ブロック書き込みパケット転送は、より高い優先度のリクエスト、例えばVRAMカラーレジスタを変え得るホストアクセスまたは他のブロック書き込みパケット転送によってインタラプトできるので、ブロック書き込みパケット転送が再開するときはいつも、ロードカラーレジスタサイクルも実行しなければならない。従って4×または8×ブロック書き込みパケット転送が始まる時、一時中断からブロック書き込みパケット転送が再開するとき、ホストが画像システムバスを使用し、これを戻した後ブロック書き込みパケット転送が続くときはいつも、ロードカラーレジスタが始まる。アクセスされているメモリがシミュレートされたブロック書き込みを必要とする場合には、ロードカラーレジスタサイクルは実行されない。

【0392】一旦カラーラッチがロードされると、上記条件のうちの1つが発生しなければ、別のロードカラーレジスタサイクルを実行することはない。例えば8×モードでブロック書き込みが開始し、次にシミュレートされたモードと8×モードが交互に変わると、8×モードとなるたびにロードカラーレジスタが繰り返されることはない。

【0393】ブロック書き込みパケット転送のための事象シーケンスは次のとおりである。転送コントローラ80は、実行すべきロードカラーレジスタステータスコードおよび第1ブロック書き込みのアドレスを出力する。次に転送コントローラ80は、BS[1:0]ピンに入力された値を読み出す。BS[1:0]が10または11に等しければ、パケット転送パラメータ内に含まれる64ビットのカラーレジスタ値も用いてロードカラーレジスタサイクルを完了する。この後に、4×または8×ブロックサイクルを発生してパケット転送を完了する。一方、BS[1:0]が00であれば、ロードカラーレジスタサイクルはデータとしてパケット転送パラメータ内に含まれる64ビットのカラーレジスタ値およびバイト選択信号としてソースデータビットを用いる通常のページモードの書き込みとなる。ページ変更またはより高い優先度のサイクルからのインタラプトのために、新しい行アクセスが開始された場合、実行すべきロードカラーレジスタのステータスコードおよび次のブロック書き込みのアドレスを出力する工程を、次の宛て先アドレスに対して繰り返す。

【0394】ロードカラーレジスタサイクルが一旦実行されると、次のようにシーケンスが続く。転送コントローラ80がブロック書き込みステータスコードおよび実

行すべき次のブロック書き込みのアドレスを出力する。BS [1:0] が10または11である場合、ソースデータビットを用いてブロック書き込みページモードサイクルが完了される。新しい行アクセスが開始されると、ブロック書き込みステータスコードおよび実行すべき次のブロック書き込みのアドレスを出力するステップが繰り返される。他方、BS [1:0] が00であれば、ブロック書き込みサイクルはパケット転送内に含まれる64ビットのカラーレジスタ値をデータとして使用し、ソースデータビットをバイト選択ビットとして使用する、通常のページモード書き込みとなる。新しい行アクセスが開始されると、ブロック書き込みステータスコードおよび実行すべき次のブロック書き込みのアドレスを出力するステップが次の宛て先アドレスに対して繰り返される。

【0395】転送コントローラ80は実際の4×または8×ブロック書き込みを常に実行するように試みる。従ってシミュレートされたブロック書き込みモード中に生じる通常の書き込みサイクルは、常にロードカラーレジスタまたはブロック書き込みステータスコードを有する。

【0396】パケット転送オプションフィールド内のパケットアクセスモードが、シリアルレジスタ転送モードを選択する011であれば、転送コントローラ80のデータ転送モードがディスエーブルされる。転送コントローラ80はアドレスおよびCAS' を出力するだけである。転送コントローラは更に行時間にTRG'、W' およびDSF [1:0] ピンもドライブし、読み出し転送または書き込み転送のVRAMモードのいずれかを選択する。これらオペレーションのいずれかを実行する際、D [63:0] ピンは行時間に16進数FFFFFFFFFFFFFFFFのマスク値を出力して、VRAM転送マスクをディスエーブルする。

【0397】メモリからレジスタへの転送である読み出し転送を実行するためのアドレスを発生するのにソースパラメータが使用され、レジスタからメモリへの転送である書き込み転送を実行するのに宛て先メモリが使用される。

【0398】ソースおよび宛て先の双方によって実行される各アクセスは、1回の行アクセスである。転送コントローラ80を介するデータの転送は行われないので、宛て先アクセス前にすべてのソースアクセスが実行される。各ソースアクセスはVRAMの行をVRAMシフトレジスタ内に転送させる。実際にはソースパラメータは通常整列したアドレス、1のAカウント、0のBカウントおよび0のCカウントでセットアップされる。従って1回の転送しか実行されない。読み出し転送に対してはすべての反転CASラインはアクティブであるので、ソースバスサイズ以下のAカウントをセッティングする結果、1回の転送が行われる。従ってほとんどの状況に対

して1のソースAカウントが理想的である。

【0399】各宛て先アクセスは、VRAMシフトレジスタのデータをVRAMメモリアレイの行内に転送させる。通常、各宛て先アクセスは次の逐次行アドレスに対する整列された転送となっている。これは、整列された開始アドレス、1のAカウント、行数-1のBカウントおよびVRAM行アドレスピッチに等しいBピッチで宛て先パラメータをセットアップすることによって行われる。書き込み転送中、すべての反転CASラインが附勢されるので、宛て先バスサイズ以下のAカウントをセットする結果、ライン当たり1回の転送が行われる。従って1の宛て先Aカウントはほとんどの状況に対して理想的である。

【0400】シフトレジスタ転送パケット転送は、VRAMシフトレジスタの内容を変え得る優先度のより高いリクエストによりインタラプトされ得るので、VRAMシフトレジスタが損なわれる可能性がある場合はいつも、読み出し転送サイクルを実行する必要がある。従って、シフトレジスタ転送パケット転送が開始するとき、シフトレジスタ転送パケット転送が一時中断された後に再開するとき、更にホストが使用され、画像システムバスに復帰した後にシフトレジスタ転送パケット転送が続くときはいつも、シフトレジスタ転送パケット転送ノソースオペレーション、すなわち読み出し転送が行われる。

【0401】図37にシリアルレジスタ転送パケット転送の簡単な例が示されている。行0は所望のパターンにセットされているものとする。パケット転送はこのパターンは行2、4、6、8および10にコピーし、ストライプ状効果を発生するためのものである。

【0402】パケット転送オプションフィールド内のパケット転送アクセスモードビットを、1XXにセットすることにより、トランスペアレンシーモードをイネーブルする。トランスペアレンシーモードのうちの1つを指定することによりソース上でのトランスペアレンシーオペレーションがイネーブルされる。ソースデータはパケット転送パラメータ内に指定された64ビットのトランスペアレンシーモード値と比較される。トランスペアレンシーは8、16、32または64ビットデータサイズとして指定できる。1回の64ビットの比較、2回の32ビットの比較、4回の64ビットの比較または8回の8ビットの比較を行う。比較の結果が真であれば、トランスペアレンシーおよびバイト書き込み回路353 (図38に示される) が対応するバイトストローブをディスエーブルし、宛て先バイトの書き込みを防止する。トランスペアレンシーはオフチップの宛て先にしかサポートされていない。オンチップの宛て先に対するトランスペアレンシーを指定すると、エラー条件によりパケット転送が一時中断される。

【0403】ソースデータが宛て先および外部バスサイ

ズに整列された後、トランスペアレンシーおよびバイト書き込み回路353(図38に示されている)によりトランスペアレンシー検出が行われる。バスサイズが64ビット未満であっても、データのうちのすべての8バイトがトランスペアレンシー値の対応する8バイトと比較される。8回の比較は、トランスペアレンシーデータサイズに従ってグループ分けされる。グループ内の比較されるバイトがすべて一致すると、そのグループに関連するバイトストロブCAS' 信号がディスエーブルされ、そのグループ内のバイトのいずれかへの書き込みが防止される。

【0404】図37a、37b、37cおよび37dは、それぞれ64ビット、32ビット、16ビットおよび8ビットのトランスペアレンシーデータサイズに対してどのように比較を行うかを示している。&記号はグループを形成するのに、どのバイトの比較をAND演算するかを示している。図37aが示すように、64ビットのトランスペアレンシーサイズは1回の64ビットの比較を行う。ソースデータおよびトランスペアレンシー値が等しければCAS' ストロブのすべてがディスエーブルされる。それ以外の場合、8バイトのすべてが書き込まれる。図37bは、32ビットのトランスペアレンシーサイズで2回の32ビットの比較が行われ、それぞれCAS' [7:4] およびCAS' [3:0] を制御することを示している。図37cは、16ビットのトランスペアレンシーサイズにおいて、4回の16ビットの比較が行われ、それぞれCAS' [7:6]、CAS' [5:4]、CAS' [3:2] およびCAS' [1:0] を制御することを示している。図37dは8ビットのトランスペアレンシーサイズにて8回の8ビットの比較を別々に行い、CAS' [7]、CAS' [6]、CAS' [5]、CAS' [4]、CAS' [3]、CAS' [2]、CAS' [3:2]、CAS' [1:0] を制御することを示している。CAS' [7:0] ストロブは、オペレーションのエンディアンにかかわらず、データバス上の同一ビットにより常時識別される。

【0405】外部バスへの整列後、トランスペアレンシーの比較が行われる。従って外部バスサイズが32ビットであれば、64ビットのトランスペアレンシーモードが選択されていても、データは常にトランスペアレンシー値のうちのビット31~0(大エンディアンモードではビット63~32)と比較される。このトランスペアレンシー機構は整数の比較グループに分割できる現在のバスサイズで作動するようになっていく。従ってバスサイズは常にトランスペアレンシーサイズ以上でなければならない。

【0406】図38は、宛て先マルチプレクサおよび整列ロジック350の一部である、トランスペアレンシーおよびバイト書き込みロジック353の構造を示す。トランスペアレンシーレジスタ601は、パケット転送バ

ラメータからの64ビットのトランスペアレンシー値を記憶する。一連の8ビットのコンパレータ611、612、613、614、615、616、617および618は、トランスペアレンシーレジスタ601内に記憶されたトランスペアレンシー値のバイトの個々のビットと、宛て先マルチプレクサ351からの整列された宛て先データのバイトの対応するビットとを比較する。各コンパレータ611、612、613、614、615、616、617、618は、対応するビットが同じであるかどうかを表示するバイトの等しい信号を発生する。ハーフワードANDゲート621、622、623および624の一组は、バイト0と1、バイト2と3、バイト4と5、並びにバイト6と7に対する対応するハーフワードの等しい信号をそれぞれ形成する。2つのワードANDゲート631および632は、ハーフワードANDゲート621、622、623および624に接続されており、バイト0~3およびバイト4~7に対する対応するワードの等しい信号を形成する。最後に、ダブルワードのANDゲート641はワードゲート631および632に接続されており、すべての64ビットに対するダブルワードの等しい信号を形成する。

【0407】マルチプレクサ645は8個のバイトの等しい信号、4つのハーフワードの等しい信号、2つのワードの等しい信号およびダブルワードの等しい信号を受ける。このマルチプレクサ645はパケット転送オプションフィールドのパケット転送アクセスモードのビット17~16に従って選択されたトランスペアレンシーサイズの表示も受ける。これらビットは表19に示されるようにコード化されている。マルチプレクサ645の出力バイト書き込みストロブは、それぞれの入力およびトランスペアレンシーサイズに基づき、反転CAS信号を実際に発生する。8ビットのトランスペアレンシーサイズが選択される場合、バイトの等しい信号がそれぞれのバイト書き込みストロブを制御する。トランスペアレンシーカラー値と、対応するデータバイトが等しいことの表示は、バイト書き込みストロブを禁止する。従ってメモリにはデータは書き込まれない。16ビットのトランスペアレンシーサイズが選択されると、各ハーフワードの等しい信号は2つの対応するバイト書き込みストロブを制御し、よって、各ハーフワードの等しい信号は2バイトの書き込みストロブを制御する。32ビットのトランスペアレンシーサイズが制御されると、各ワードの等しい信号は4つの対応するバイト書き込みストロブを制御する。64ビットのトランスペアレンシーサイズが選択されると、8つのバイト書き込みストロブのすべてがダブルワードの等しい信号によって制御される。従って、選択されたトランスペアレンシーサイズに基づき、トランスペアレンシーサイズに等しいデータ部分とトランスペアレンシーカラー値との同一性により、データの宛て先への書き込みがアボートされる。こ

の機能は多くのグラフィックアプリケーションで極めて有効である。

【0408】パケット転送オプションのパケットアクセスモードが001に等しいときに、周辺デバイスモードが選択されると、周辺デバイス転送が実行される。このモードは周辺デバイスが転送コントローラ80のメモリコントローラを活用し、マルチプロセッサ集積回路100の外部のメモリとの間における読み出しまたは書き込みを可能にする。周辺デバイス転送が行われると、転送コントローラ80はメモリアドレスおよび制御ラインをドライブするが、データを読み出ししたり、ドライブ出力させることはない。これにより、周辺デバイス、例えば画像システムバスに接続されているホストシステム1は、データの書き込みまたは読み出しを可能にする。周辺転送の読み出しまたは書き込み方向は、パケット転送パラメータをプログラムする方法により決定される。

【0409】メモリから周辺デバイスへの読み出しを行う周辺読み出し転送は、周辺デバイスが必要とするメモリデータにアクセスするよう、パケット転送のソースパラメータをプログラムすることによって発生される。ソースアクセスモードは、次元の定められた転送またはガイド転送のいずれかにできるが、値でフィルする転送にはできない。宛て先転送は宛て先転送モードを000フィールドにセットし、宛て先Aカウントを0にセットすることによりディスエーブルしなければならない。

【0410】周辺デバイスからメモリへの書き込みを行う周辺書き込み転送は、周辺デバイスが書き込みを必要とするメモリエリアにアクセスするよう、宛て先パラメータをプログラムすることによって発生される。宛て先アクセスモードは次元の定められた転送またはガイド転送のいずれかにできる。ソース転送はソース転送モードを000にセットし、ソースAカウントを0にセットすることによりディスエーブルしなければならない。バイトの宛て先番号がバイトのソース番号を越えたとしても、この例ではパケット転送エラーは生じない。

【0411】周辺デバイスパケット転送リクエストは、任意の優先度のプロセッサにより送信でき、通常の優先度決定方法を用いてサービスされる。しかしながら通常のオペレーションはデータの読み出し、書き込みを望む際に、周辺デバイスに転送を開始させるようになっていく。このような開始は、外部インタラプトを用いることによって行われる。周辺デバイスがサービスを必要とする際、周辺デバイスはマルチプロセッサ集積回路100の外部のインタラプト入力の一つにより、マルチプロセッサ60をインタラプトできる。マスタプロセッサ60は次に、インタラプトサービスルーチンの一部として周辺デバイスパケット転送を送ることができる。サービスルーチン内でインタラプトがディスエーブルされれば、この転送は緊急優先度を送ることにより、デジタル画像／グラフィックプロセッサ71、72、73、74のキ

ャッシュリクエストよりも高い優先度を与えることができる。

【0412】周辺デバイスがマスタプロセッサ60を一旦インタラプトすると、周辺デバイスは転送コントローラ80がデータの読み出し、書き込みをできる前に、パケット転送を開始するまで待機しなければならない。周辺デバイスの転送の開始は、行時間におけるステータス〔4:0〕上に出力される特別サイクルタイプのコードによって信号が送られる。読み出しに対しては値00100が使用され、書き込みに対しては値00101が使用される。周辺デバイス転送の開始点および終了点は、常にLASTPAGEレジスタ360を無効にし、行アクセスを強制的に行わせる。周辺デバイスはデータ転送をいつ行うかを決定するよう、STATUS〔4:0〕をモニタしなければならない。転送コントローラ80により発生されるメモリサイクルのバスサイズのタイプ、列タイミング等は、サイクルの開始点におけるメモリ識別入力バスサイズ選択BS〔1:0〕、列タイミング選択CT〔1:0〕等によって選択されるものに対応する。次に周辺デバイスは、CAS'〔7:0〕、CLKOUT等を用いる、その後の列アクセスにそのデータ転送を同期化できる。

【0413】メモリのアドレス指定はパケット転送パラメータによって行われるので、転送周辺デバイスでは転送コントローラ80によってアクセスされる順で、データを送受信するよう準備が整っていなければならない。周辺デバイスは転送コントローラ80の転送レートに合致したり、転送を低速にするよう、待機ステートを挿入できるようになっていなければならない。

【0414】周辺デバイスは転送のために画像システムバスを使用しているので、周辺デバイス転送が開始するまでバスをドライブしてはならない。これは周辺デバイスとマルチプロセッサ集積回路100のデータバスとの間にトランシーバを置き、周辺デバイス転送中にこれらをイネーブルするだけで達成できる。マルチプロセッサ集積回路100は転送中にそのデータバスを高インピーダンスとし、外部トランシーバが設けられていれば、これをディスエーブルするよう、反転DBENを高レベルで非アクティブとなるようにドライブする。

【0415】転送コントローラ80は、大エンディアンフォーマットまたは小エンディアンフォーマットのいずれかでデータにアクセスできる。このエンディアンモードは、バイトにアクセスする方法を選択する。小エンディアンフォーマットではバイト0はワード内の最も右側のバイトであり、その後続くバイトは左に向かって番号がつけられる。大エンディアンフォーマットではバイト0はワード内の最も左側のバイトであり、その後続くバイト2は右側に番号がつけられる。

【0416】アドレスの最小位の3ビットおよび転送すべきバイト数は、有効データバイトの位置を決定する。

表 2 4 a および 2 4 b は、小エンディアンモードで 6 4 * ことを示す。
 ビットバス転送のためのバイト位置を示す。表 2 4 a および 2 4 b では V は有効バイト位置を示し、0 は無効バイト位置を示す。点線はそのオペレーションを実行できない*

【0 4 1 7】

【表 2 4】

3つの最小位 のアドレス ビット	バ イ ト 数			
	1バイト	2バイト	3バイト	4バイト
0 0 0	0000000V	000000VV	00000VVV	0000VVVV
0 0 1	000000V0	00000VV0	0000VVV0	000VVVV0
0 1 0	00000V00	0000VV00	000VVV00	00VVVV00
0 1 1	0000V000	000VV000	00VVV000	0VVVV000
1 0 0	000V0000	00VV0000	0VVV0000	VVVV0000
1 0 1	00V00000	0VV00000	VVV00000	--
1 1 0	0V000000	VV000000	--	--
1 1 1	V0000000	--	--	--

表 2 4 a

3つの最小位 のアドレス ビット	バ イ ト 数			
	5バイト	6バイト	7バイト	8バイト
0 0 0	000VVVVV	00VVVVVV	0VVVVVVV	VVVVVVVV
0 0 1	00VVVVV0	0VVVVVV0	VVVVVVV0	--
0 1 0	0VVVVVV0	VVVVVVV0	--	--
0 1 1	VVVVVV00	--	--	--
1 0 0	--	--	--	--
1 0 1	--	--	--	--
1 1 0	--	--	--	--
1 1 1	--	--	--	--

表 2 4 b

【0 4 1 8】表 2 5 a および 2 5 b は、大エンディアン * 【0 4 1 9】
 モードにおける 6 4 ビットバス転送のためのバイト位置 【表 2 5】
 を示す。

※

3つの最小位 のアドレス ビット	バ イ ト 数			
	1バイト	2バイト	3バイト	4バイト
0 0 0	V0000000	VV000000	VVV00000	VVVV0000
0 0 1	0V000000	0VV00000	0VVV0000	0VVVV000
0 1 0	00V00000	00VV0000	00VVV000	00VVVV00
0 1 1	000V0000	000VV000	000VVV00	000VVVV0
1 0 0	0000V000	0000VV00	0000VVV0	0000VVVV
1 0 1	00000V00	00000VV0	00000VVV	--
1 1 0	000000V0	000000VV	--	--
1 1 1	0000000V	--	--	--

表 2 5 a

3つの最小位 のアドレス ビット	バ イ ト 数			
	5バイト	6バイト	7バイト	8バイト
0 0 0	VVVVVV00	VVVVVVV0	VVVVVVV0	VVVVVVVV
0 0 1	0VVVVVV0	0VVVVVV0	0VVVVVVV	--
0 1 0	00VVVVV0	00VVVVVV	--	--
0 1 1	000VVVVV	--	--	--
1 0 0	--	--	--	--
1 0 1	--	--	--	--
1 1 0	--	--	--	--
1 1 1	--	--	--	--

表 2 5 b

【0 4 2 0】外部バスが 3 2 ビットに制限されていると
 き、データの転送にはバスのうちの最小位の 4 バイトし
 か使用しない。このことは、小エンディアン転送では D

【3 1 : 0】が使用され、大エンディアン転送に対して
 は D 【6 3 : 3 2】が使用されることを意味している。
 表 2 6 には、小エンディアンのためのアドレスの最小位

の2ビットに基づくバイト位置が示されている。Xは無
視する64ビットバスのバイトを表示し、点線は実行で
きない転送を示している。

2つの最小位 のアドレス ビット	* バ イ ト 数			
	1バイト	2バイト	3バイト	4バイト
0 0	XXXX000V	XXXX00VV	XXXX0VVV	XXXXVVVV
0 1	XXXX00V0	XXXX0VV0	XXXXVVV0	--
1 0	XXXX0V00	XXXXVV00	--	--
1 1	XXXXV000	--	--	--

表26

【0422】表27には、大エンディアンのためのアド
レスの最小位の2ビットに基づくバイト位置が示されて
いる。

2つの最小位 のアドレス ビット	※ バ イ ト 数			
	1バイト	2バイト	3バイト	4バイト
0 0	000VXXXX	00VVXXXX	0VVVXXXX	VVVVXXXX
0 1	00V0XXXX	0VV0XXXX	VVV0XXXX	--
1 0	0V00XXXX	VV00XXXX	--	--
1 1	V000XXXX	--	--	--

表27

【0424】外部バスが16ビットに制限されている
と、データの転送のために小エンディアンのための最小
位の2バイトD〔15:0〕または大エンディアンのた
めのD〔63:48〕が使用される。表28は、アドレ
スの最小位バイトに基づく小エンディアンモードのため
のバイト位置を示す。

【0425】

【表28】

最小位の アドレス ビット	バ イ ト 数	
	1バイト	2バイト
0	XXXXXXXX0V	XXXXXXXXVV
1	XXXXXXXXV0	--

表28

【0426】表29は、アドレスの最小位ビットに基づ
く小エンディアンモードのためのバイト位置を示す。

【0427】

【表29】

最小位の アドレス ビット	バ イ ト 数	
	1バイト	2バイト
0	V0XXXXXXXX	VVXXXXXXXX
1	0VXXXXXXXX	--

表29

【0428】外部バスが8ビット用に構成されていると
き、データ転送のために小エンディアン用の最小位のバ
イトD〔7:0〕または大エンディアン用D〔63:5

6〕だけが使用される。

【0429】内部クロスバー上で32ビットの外部デー
タを転送する際、転送コントローラ80は表24a、2
4b、25aおよび25bからの1バイト、2バイト、
3バイトおよび4バイトのバイト位置を用いて、通常6
4ビットのアクセスを実行する。内部クロスバー上で1
6ビットの外部データを転送する際、転送コントローラ
80は表24a、24b、25aおよび25bからの1
バイトおよび2バイトのバイト位置を用いて、通常の6
4ビットアクセスを実行する。

【0430】反転UTIME入力を用いてリセット時に
マルチプロセッサ集積回路100のエンディアンモード
が選択される。マルチプロセッサ集積回路100反転リ
セット入力上の立ち上がりエッジの前のクロックサイク
ルにおけるUTIME'の値をサンプリングし、合致す
る。UTIME'がリセットの終了時に低レベル(0)
にサンプリングされた場合、マルチプロセッサ集積回路
100は次のハードウェアのリセットが生じるまで、大
エンディアンモードで作動する。UTIME'が高レベ
ル(1)でサンプリングされると、マルチプロセッサ集
積回路100は小エンディアンモードで作動する。

【0431】パケット転送パラメータはこれまで述べた
ように、ワード(32ビット)レベルのみにおいて、エン
ディアンと独立している。転送コントローラ80はパ
ケット転送パラメータをダブルワード(64ビット)転
送として常にフェッチし、記憶する。転送コントローラ
80は選択されたエンディアンに従って32ビットのワ
ードをスワップする。ワード内の16ビットの量、例え
ばAカウントおよびBカウントは、これら値を含むフィ

ールドが単一の32ビットの量とみなされるので、エンディアンに従ってスワップされることはない。同様に、64ビットのトランスペアレンシーワードおよび64ビットのカラーレジスタ値のフィールドは、常に単一の64ビット量として取り扱われ、そのバイトはエンディアンに従ってスワップされることはない。

【0432】ローカルメモリの読み出しおよび書き込みサイクルは、メモリとプロセッサ集積回路100との間でデータとインストラクションとを転送するのに使用される。これらサイクルはパケット転送、キャッシュリクエストまたは転送コントローラ80に対するダイレクト外部アクセスリクエストの結果として生じ得る。読み出しサイクルはデータをメモリからマルチプロセッサ集積回路100へ転送する。転送コントローラ80はこのサイクルの開始点でSTATUS[4:0]上に0000を出力し、読み出しが行われていることを表示する。サイクル中WE'は高レベルで非アクティブに保持され、TRG'はRAS'の降下後に低レベルにドライブされ、メモリ出力ドライバをイネーブルし、DDIN'はデータトランシーバがマルチプロセッサ集積回路100の内部をドライブするように、このサイクル中に低レベルでアクティブとなる。転送コントローラ80はD[63:0]をメモリから駆動できるように高インピーダンスにスイッチングし、適当な熱ステートの間に入力データをラッチする。転送コントローラ80は常に64ビットのダブルワードを読み出し、次に適当なデータバイトとを抽出し、整列する。従って64ビット未満のバスサイズに対して、無効バイトを放棄する。

【0433】図39は、高品位テレビシステムにおける、本発明に係わるマルチプロセッサ集積回路100の使用法を示す。図39は、高品位テレビ信号の3つのソースを示している。これら信号としては、放送テレビ信号、コンパクトディスクのリードオンリーメモリ信号およびケーブルテレビ信号がある。

【0434】アンテナ801は高品位テレビ信号を含む放送用無線周波数信号を受信し、テレビチューナー802は特定の無線周波数信号を選択する同調受信機と、無線周波数信号上にエンコードされた画像データを抽出する複合器と、アナログ/デジタルコンバータを含む。従ってテレビチューナー802は、高品位テレビ画像に対応するデジタル信号を発生する。これらデジタル信号はバッファ803に一時的に記憶されるようになっている。

【0435】コンパクトディスクリードオンリーメモリ(CDROM)プレーヤー811は、コンパクトディスク上に永久記録されたデータを読み出す。これらデータは、所望の高品位テレビプログラムに対応した画像データを含む。コンパクトディスクリードオンリーメモリプレーヤー811は、コンパクトディスクから読み出したデジタルデータを一時記憶のためバッファ812へ供給

する。バッファ812は、画像システムバスにも接続されている。

【0436】ケーブルシステムボックス821は、ケーブルシステムに双方向に接続する。このような双方向の接続によりケーブルシステムからユーザーに高品位テレビ信号の送信を行い、更にユーザーからケーブルシステムへリクエスト、質問等の送信を行うことができるようになっている。ケーブルシステムはユーザーへのデジタル送信を利用したり、上記無線周波数放送に類似する無線周波数送信を利用したりできる。ケーブルシステムボックス821は、デジタル画像データを一時記憶できるよう、バッファ822に供給するための必要な変換回路を含む。ここでバッファ822画像システムバスにも接続されていることに留意されたい。高品位テレビは送信モードに拘わらずデータ圧縮フォーマットで送信する可能性がかなり高い。マイクロプロセッサ集積回路100は、圧縮されたデータを受信し、このデータを個々のテレビフレームにデコンプレス(圧縮解凍)し、フレームデータをビデオランダムアクセスメモリ6に供給するようにプログラムされている。先に述べたように、このデータはビデオランダムアクセスメモリ6からリコールされ、ビデオバレット7へ供給される。ビデオバレット7は、適当なビデオ信号を発生し、ビデオディスプレイ8をドライブする。画像データをディスプレイに供給する際に、画像データをデコンプレスする方法は、マイクロプロセッサ集積回路100内および画像システムバスに沿った多数のデータ移動を伴う。転送コントローラ80はマスタプロセッサ60およびデジタル画像/グラフィックプロセッサ71、72、73および74からのパケット転送用リクエストに応答し、このデータ移動を制御し、更にメモリリフレッシュのような他の画像システムバスの使用と、このデータ移動とを調和させる。

【0437】図40は、本発明の別のシステムの実施例を示す。図40では、マルチプロセッサの集積回路101はマスタプロセッサ60と単一のデジタル画像/グラフィックプロセッサ71を含む。マルチプロセッサ集積回路101はマルチプロセッサ集積回路100よりも狭いシリコン基板面積しか必要としないので、より安価に製造できる。マルチプロセッサ集積回路101は、マルチプロセッサ集積回路100の製造に対して先に述べた技術と同じ技術を用いて製造される。各デジタル画像/グラフィックプロセッサの幅は、対応するメモリおよびクロスバー50の関連する部分の幅と一致するので、マルチプロセッサ集積回路100をデジタル画像/グラフィックプロセッサ71と72との間でカットし、マルチプロセッサ集積回路101を得ることができる。4つのデジタル画像/グラフィックプロセッサの処理容量が不要の場合には、アプリケーションのためマルチプロセッサ集積回路101を用いることができる。

【0438】図42では、マルチプロセッサ集積回路1

01がカラーファクシミリ装置の一部として示されている。モデム1301は送受信のための電話回線に双方に結合されている。モデム1301は、バッファ1302とも通信し、このバッファは画像システムバスに更に結合されている。モデム1301は電話回線を介してファクシミリ信号を受信し、モデム1301はこれら信号を復調し、復調信号は次にバッファ1302に一時的に記憶される。転送コントローラ80はデジタル画像/グラフィックプロセッサ71によって処理できるよう、データメモリ22、23、24へデータを転送することにより、バッファ1302にサービスする。デジタル画像/グラフィックプロセッサ71が、入進データの前に位置づけることができない場合、転送するコントローラ80はこのデータの場合1302からメモリ9へ転送することもできる。デジタル画像/グラフィックプロセッサ71は入進ファクシミリの画像データを処理する。この処理では、画像デコンプレッション、ノイズ低減、誤り訂正、カラーベース補正等を行ってもよい。一旦処理した場合、転送コントローラ80は画像データをデータメモリ22、23、24からビデオランダムアクセスメモリ(VRAM)1303へ転送する。プリンタコントローラ1304は、フレームコントローラ90の制御により、画像データをリコールし、これをカラープリンタ1305へ供給し、このプリンタはハードコピーを作成する。

【0439】図40の装置はカラーファクシミリを送ることもできる。撮像デバイス3はソース原画をスキャンする。撮像デバイス3はフレームコントローラ90の制御により作動している画像キャプチャコントローラ4へ、生の画像データを供給する。この画像データは、ビデオランダムアクセスメモリ1303に記憶される。図40に示された実施例は、別個のビデオランダムアクセスメモリを利用している図1の実施例と対照的に、画像キャプチャと画像ディスプレイの双方のためにビデオランダムアクセスメモリ1303を共用している。転送コントローラ80は、この画像データをデータメモリ22、23、24へ転送する。次にデジタル画像/グラフィックプロセッサ71は、データ圧縮、誤り訂正冗長性、カラーベース補正等のために画像データを処理する。この処理されたデータはファクシミリ転送をサポートするのに必要のように、転送コントローラ80によってバッファ1303へ転送される。転送コントローラ80は相対的データレートに応じてバッファ1302への転送前に一時的にメモリ9にデータを記憶する。バッファ1302内のこの画像データは、モデム1301により変調され、電話回線を通して送信される。

【0440】撮像デバイスとカラープリンタとが同じシステム内に設けられているので、このシステムはカラー複写機としても作動できることに留意されたい。この場合、データ圧縮とデコンプレッションは不要である。し

かしながらノイズ低減およびカラーベース補正のためには、まだデジタル画像/グラフィックプロセッサ71が有効である。コピーが原画と異なるカラーを有するように、色を注意深くずらすように、デジタル画像/グラフィックプロセッサ71をプログラムすることも可能である。フォールスカラーリングとして知られているこの技術は、データのダイナミックレンジを利用可能なプリントカラーのダイナミックレンジに合わせるのに有効である。

【0441】以上の説明に関して更に以下の項を開示する。

(1) 複数の対応するアドレスにデータを記憶するメモリと、パケット転送リクエストと、パケット転送パラメータとを受ける動作をする制御回路であって、該パケット転送パラメータが、スタートアドレスと、ある数のガイドテーブルエントリと、テーブルポインタとを含む、前記制御回路と、前記数のガイドテーブルエントリを含むガイドテーブルであって、それぞれのガイドテーブルエントリが、アドレス値およびアドレスのブロックを定義するディメンション値を含み、前記テーブルポインタが最初に前記ガイドテーブル内の第1ガイドテーブルエントリをポインタする、前記ガイドテーブルと、前記制御回路に結合せしめられたアドレス発生回路であって、該アドレス発生回路が、前記パケット転送パラメータから、前記スタートアドレスと、前記数のガイドテーブルエントリと、前記テーブルポインタとを受ける動作をし、前記アドレス発生回路が、前記ガイドテーブルエントリに対応するメモリアccess用のアドレスのブロックの集合を、前記スタートアドレスと、前記テーブルポインタによりポインタされた前記ガイドテーブルエントリの前記アドレス値と、の所定の組合せから、ブロックスタートアドレスであって、該第1ブロックスタートアドレスが前記スタートアドレスである、前記ブロックスタートアドレスを形成し、該ブロックスタートアドレスと、前記テーブルポインタによりポインタされた前記ガイドテーブルエントリの前記ディメンション値と、からアドレスのブロックを形成し、アドレスの該ブロックにメモリアccessを行い、もしガイドテーブルエントリの前記数よりも少ないメモリアccessしか行われ終わっていないなければ、前記メモリアccessに続いて、前記ガイドテーブルの次のエントリをポインタするように前記テーブルポインタを更新する、ことによって形成する、前記アドレス発生回路と、を含む、データ処理装置。

【0442】(2) 前記アドレス発生回路が、前記アドレス値を前記前のブロックスタートアドレスに加算することにより、前記スタートアドレスと、前記テーブルポインタによりポインタされた前記ガイドテーブルエントリの前記アドレス値と、の前記所定の組合せを形成する、第1項記載のデータ処理装置。

【0443】(3) 前記アドレス発生回路が、前記テー

ブルポイントによりポイントされた前記ガイドテーブル値を前記スタートアドレスに加算することにより、前記スタートアドレスと、前記テーブルポイントによりポイントされた前記ガイドテーブルエントリの前記アドレス値と、の前記所定の組合せを形成する、第1項記載のデータ処理装置。

【0444】(4) それぞれのガイドテーブルエントリの前記ディメンション値が、画素のアレイの水平ディメンション値および垂直ディメンション値を含む、第1項記載のデータ処理装置。

【0445】(5) 前記メモリアクセスが、アドレスの前記ブロックからのメモリアドレスから成る、第1項記載のデータ処理装置。

【0446】(6) 前記メモリアクセスが、アドレスの前記ブロックへのメモリアドレスから成る、第1項記載のデータ処理装置。

【0447】(7) 単一半導体チップ上に、メモリと、パケット転送リクエストを発生する回路を有するデータプロセッサと、前記メモリと、前記データプロセッサと、に接続されたデータ転送制御装置であって、該データ転送制御装置が、パケット転送リクエストと、パケット転送パラメータとを受ける動作をする制御回路であって、該パケット転送パラメータが、スタートアドレスと、ある数のガイドテーブルエントリと、テーブルポイントとを含む、前記制御回路と、前記数のガイドテーブルエントリを含むガイドテーブルであって、それぞれのガイドテーブルエントリが、アドレス値およびアドレスのブロックを定義するディメンション値を含み、前記テーブルポイントが最初に前記ガイドテーブル内の第1ガイドテーブルエントリをポイントする、前記ガイドテーブルと、前記制御回路に結合せしめられたアドレス発生回路であって、該アドレス発生回路が、前記パケット転送パラメータから、前記スタートアドレスと、前記数のガイドテーブルエントリと、前記テーブルポイントとを受ける動作をし、前記アドレス発生回路が、前記ガイドテーブルエントリに対応するメモリアクセス用のアドレスのブロックの集合を、前記スタートアドレスと、前記テーブルポイントによりポイントされた前記ガイドテーブルエントリの前記アドレス値と、の所定の組合せから、ブロックスタートアドレスであって、該第1ブロックスタートアドレスが前記スタートアドレスである、前記ブロックスタートアドレスを形成し、該ブロックスタートアドレスと、前記テーブルポイントによりポイントされた前記ガイドテーブルエントリの前記ディメンション値と、からアドレスのブロックを形成し、アドレスの該ブロックにメモリアクセスを行い、もしガイドテーブルエントリの前記数よりも少ないメモリアクセスしか行われ終わっていなければ、前記メモリアクセスに続いて、前記ガイドテーブルの次のエントリをポイントするように前記テーブルポイントを更新する、ことによつて

形成する、前記アドレス発生回路と、を含む、前記データ転送制御装置と、を備えた、集積回路。

【0448】(8) 前記アドレス発生回路が、前記アドレス値を前記前のブロックスタートアドレスに加算することにより、前記スタートアドレスと、前記テーブルポイントによりポイントされた前記ガイドテーブルエントリの前記アドレス値と、の前記所定の組合せを形成する、第7項記載の集積回路。

【0449】(9) 前記アドレス発生回路が、前記テーブルポイントによりポイントされた前記ガイドテーブル値を前記スタートアドレスに加算することにより、前記スタートアドレスと、前記テーブルポイントによりポイントされた前記ガイドテーブルエントリの前記アドレス値と、の前記所定の組合せを形成する、第7項記載の集積回路。

【0450】(10) それぞれのガイドテーブルエントリの前記ディメンション値が、画素のアレイの水平ディメンション値および垂直ディメンション値を含む、第7項記載の集積回路。

【0451】(11) 前記メモリアクセスが、アドレスの前記ブロックからのメモリアドレスから成る、第7項記載の集積回路。

【0452】(12) 前記メモリアクセスが、アドレスの前記ブロックへのメモリアドレスから成る、第7項記載の集積回路。

【0453】(13) 前記単一半導体チップ上に含まれない外部メモリにアクセスする動作をする外部メモリインタフェースをさらに含み、前記アドレス発生回路が、前記メモリおよび前記外部メモリの双方を含むアドレスを発生する、第7項記載の集積回路。

【0454】(14) 複数の位置に情報を記憶するステップと、パケットリクエストを受けてスタートアドレスおよび制御信号を供給するステップと、該スタートアドレス、制御信号、およびパッチ値に応答してパッチアドレスを発生するステップと、パッチ情報をデコードしてディメンション値およびパッチ値を供給するステップと、前記スタートアドレスと、前記ディメンション値と、前記パッチ値とに応答してパッチアドレスを発生するステップと、を含む、データ処理装置を操作する方法。

【0455】(15) 複数の対応するアドレスにデータを記憶するメモリと、パケット転送リクエストと、パケット転送パラメータとを受ける動作をする制御回路であって、該パケット転送パラメータが、スタートアドレスと、アドレスのブロックを定義するディメンション値と、ある数のガイドテーブルエントリと、テーブルポイントとを含む、前記制御回路と、前記数のガイドテーブルエントリを含むガイドテーブルであって、それぞれのガイドテーブルエントリがアドレス値を含み、前記テーブルポイントが最初に前記ガイドテーブル内の第1ガイ

ドテーブルエントリをポイントする、前記ガイドテーブルと、前記制御回路と、前記ガイドテーブルと、に結合せしめられたアドレス発生回路であって、該アドレス発生回路が、前記パケット転送パラメータから、前記スタートアドレスと、前記ディメンション値と、前記数のガイドテーブルエントリと、前記テーブルポインタとを受ける動作をし、前記アドレス発生回路が、前記ガイドテーブルエントリに対応するメモリアクセス用のアドレスのブロックの集合を、前記スタートアドレスと、前記テーブルポインタによりポイントされた前記ガイドテーブルエントリの前記アドレス値と、の所定の組合せから、ブロックスタートアドレスであって、該第1ブロックスタートアドレスが前記スタートアドレスである、前記ブロックスタートアドレスを形成し、該ブロックスタートアドレスと、前記ディメンション値とからアドレスのブロックを形成し、アドレスの該ブロックにメモリアクセスを行い、もしガイドテーブルエントリの前記数よりも少ないメモリアクセスしか行われ終わっていなければ、前記メモリアクセスに続いて、前記ガイドテーブルの次のエントリをポイントするように前記テーブルポインタを更新する、ことによって形成する、前記アドレス発生回路と、を含む、データ処理装置。

【0456】(16) 前記アドレス発生回路が、前記アドレス値を前記前のブロックスタートアドレスに加算することにより、前記スタートアドレスと、前記テーブルポインタによりポイントされた前記ガイドテーブルエントリの前記アドレス値と、の前記所定の組合せを形成する、第15項記載のデータ処理装置。

【0457】(17) 前記アドレス発生回路が、前記テーブルポインタによりポイントされた前記ガイドテーブル値を前記スタートアドレスに加算することにより、前記スタートアドレスと、前記テーブルポインタによりポイントされた前記ガイドテーブルエントリの前記アドレス値と、の前記所定の組合せを形成する、第15項記載のデータ処理装置。

【0458】(18) 前記パケット転送パラメータの前記ディメンション値が、画素のアレイの水平ディメンション値および垂直ディメンション値を含む、第15項記載のデータ処理装置。

【0459】(19) 前記メモリアクセスが、アドレスの前記ブロックからのメモリアクセスから成る、第15項記載のデータ処理装置。

【0460】(20) 前記メモリアクセスが、アドレスの前記ブロックへのメモリアクセスから成る、第15項記載のデータ処理装置。

【0461】(21) 単一半導体チップ上に、メモリと、パケット転送リクエストを発生する回路を有するデータプロセッサと、前記メモリと、前記データプロセッサと、に接続されたデータ転送制御装置であって、該データ転送制御装置が、パケット転送リクエストと、パケ

ット転送パラメータとを受ける動作をする制御回路であって、該パケット転送パラメータが、スタートアドレスと、アドレスのブロックを定義するディメンション値と、ある数のガイドテーブルエントリと、テーブルポインタとを含む、前記制御回路と、前記数のガイドテーブルエントリを含むガイドテーブルであって、それぞれのガイドテーブルエントリがアドレス値を含み、前記テーブルポインタが最初に前記ガイドテーブル内の第1ガイドテーブルエントリをポイントする、前記ガイドテーブルと、前記制御回路と、前記ガイドテーブルと、に結合せしめられたアドレス発生回路であって、該アドレス発生回路が、前記パケット転送パラメータから、前記スタートアドレスと、前記ディメンション値と、前記数のガイドテーブルエントリと、前記テーブルポインタとを受ける動作をし、前記アドレス発生回路が、前記ガイドテーブルエントリに対応するメモリアクセス用のアドレスのブロックの集合を、前記スタートアドレスと、前記テーブルポインタによりポイントされた前記ガイドテーブルエントリの前記アドレス値と、の所定の組合せから、ブロックスタートアドレスであって、該第1ブロックスタートアドレスが前記スタートアドレスである、前記ブロックスタートアドレスを形成し、該ブロックスタートアドレスと、前記パケット転送パラメータの前記ディメンション値とからアドレスのブロックを形成し、アドレスの該ブロックにメモリアクセスを行い、もしガイドテーブルエントリの前記数よりも少ないメモリアクセスしか行われ終わっていなければ、前記メモリアクセスに続いて、前記ガイドテーブルの次のエントリをポイントするように前記テーブルポインタを更新する、ことによって形成する、前記アドレス発生回路と、を含む、前記データ転送制御装置と、を備えた、集積回路。

【0462】(22) 前記アドレス発生回路が、前記アドレス値を前記前のブロックスタートアドレスに加算することにより、前記スタートアドレスと、前記テーブルポインタによりポイントされた前記ガイドテーブルエントリの前記アドレス値と、の前記所定の組合せを形成する、第21項記載の集積回路。

【0463】(23) 前記アドレス発生回路が、前記テーブルポインタによりポイントされた前記ガイドテーブル値を前記スタートアドレスに加算することにより、前記スタートアドレスと、前記テーブルポインタによりポイントされた前記ガイドテーブルエントリの前記アドレス値と、の前記所定の組合せを形成する、第21項記載の集積回路。

【0464】(24) 前記パケット転送パラメータの前記ディメンション値が、画素のアレイの水平ディメンション値および垂直ディメンション値を含む、第21項記載の集積回路。

【0465】(25) 前記メモリアクセスが、アドレスの前記ブロックからのメモリアクセスから成る、第21項

記載の集積回路。

【0466】(26) 前記メモリアクセスが、アドレスの前記ブロックへのメモリライトから成る、第21項記載の集積回路。

【0467】(27) 前記単一半導体チップ上に含まれない外部メモリにアクセスする動作をする外部メモリインタフェースをさらに含み、前記アドレス発生回路が、前記メモリおよび前記外部メモリの双方を含むアドレスを発生する、第21項記載の集積回路。

【0468】(28) 複数の位置に情報を記憶するステップと、パケットリクエストを受けて、スタートアドレスと、制御信号と、パッチ値と、を供給するステップと、該スタートアドレスと、該制御信号と、該パッチ値と、に応答してパッチアドレスを発生するステップと、を含む、画像処理装置を操作する方法。

【0469】(29) 本発明は、メモリアクセスのアドレスの制御様式に関する。本発明のデータ処理装置は、メモリと、制御回路と、ガイドテーブルと、アドレス発生回路と、を含む。該制御回路は、パケット転送リクエストと、パケット転送パラメータとを受ける。該パケット転送パラメータは、スタートアドレスと、ある数のガイドテーブルエントリと、テーブルポインタとを含む。前記ガイドテーブルは、ガイドテーブルエントリを含み、それぞれのガイドテーブルエントリは、アドレス値およびアドレスのブロックを定義するディメンション値を含む。前記テーブルポインタは、最初に前記ガイドテーブル内の第1ガイドテーブルエントリをポイントする。前記アドレス発生回路は、前記スタートアドレスと、前記ガイドテーブルエントリの前記アドレス値と、の所定の組合せから形成されるスタートアドレスを有する、それぞれのガイドテーブルエントリに対応するメモリアクセス用のアドレスのブロックの集合を形成する。アドレスの該ブロックは、前記ディメンション値から形成される。前記メモリアクセスに続いて、前記アドレス発生回路は、前記ガイドテーブルの次のエントリをポイントするように前記テーブルポインタを更新する。前記アドレス発生回路は、随意選択的に、前記アドレス値を前記前のブロックスタートアドレスに加算することにより、または、前記ガイドテーブル値を前記スタートアドレスに加算することにより、スタートアドレスと、ガイドテーブルエントリのアドレス値と、の前記所定の組合せを形成しうる。前記メモリアクセスは、アドレスの前記ブロックからのメモリリード、または、アドレスの前記ブロックへのメモリライトでありうる。実施例においては、メモリと、データプロセッサと、上述のメモリアクセスを行うデータ転送制御装置とは、単一半導体チップ内に構成される。該データ転送制御装置は、オンチップメモリと同様に、外部メモリにアクセスしうる。

【0470】警告：著作権1991年テキサスインスツルメンツ社 本特許文献の開示の一部は、著作権およ

びマスクワーク保護の対象となる資料を含む。この著作権およびマスクワークの所有者は、米国特許庁の特許ファイルまたは記録にある特許文献または特許の開示の、何人によるコピーに反対するものではないが、それ以外に対しては、すべての著作権およびマスクワークの権利を留保するものである。

【0471】関連出願とのクロスレファレンス：本願は、下記の米国特許および継続中の米国特許出願に開示された発明の改良に関するもので、下記の特許および米国特許出願はすべてテキサスインスツルメンツ社に譲渡されたものであり、これらのいずれも参考例として援用する。

【0472】1994年6月21日に提出された「プロセッサとメモリをクロスバーリンクしたマルチプロセッサおよびその作動方法」を発明の名称とする、米国特許出願第08/263,501号。この出願は、1993年10月12日出願され現在放棄された米国特許出願第08/135,754号の継続出願であり、1992年8月21日に提出され現在放棄されている米国特許出願第07/933,865号の継続出願であり、この米国特許出願第07/933,865号は、1989年11月17日出願され現在放棄されている米国特許出願第07/435,591号の継続出願である。

【0473】1989年11月17日出願され1993年5月18日に発行された「SIMD/MIMD再構成可能なマルチプロセッサおよびオペレーション方法」を発明の名称とする米国特許第5,212,777号。

【0474】1989年11月17日出願され現在放棄されている米国特許出願第07/437,856号の継続出願である、1992年6月5日出願され現在放棄されている米国特許出願第07/895,565号の継続出願である、1994年6月22日出願された「マルチプロセッサ用再構成可能な通信およびそのオペレーション方法」を発明の名称とする米国特許出願第08/264,111号。

【0475】1989年11月17日出願され現在放棄されている米国特許出願第07/437,852号の継続出願である、1994年6月22日出願された「小エリアのクロスバーおよびそのオペレーション方法」を発明の名称とする米国特許出願第08/264,582号。

【0476】1989年11月17日出願され現在放棄されている米国特許出願第07/437,853号の継続出願である、1993年5月15日出願された「同期されたMIMDマルチ処理システムおよびそのオペレーション方法」を発明の名称とする米国特許出願第08/032,530号。

【0477】1989年11月17日出願され、1993年3月23日発行された「スライスされたアドレス指定用マルチプロセッサおよびそのオペレーション方法」

を発明の名称とする米国特許第5, 197, 140号。

【0478】1989年11月17日出願され、1994年8月16日発行された「画像データの二進ストリーム内の1の数をカウントするための相互接続された半加算器のマトリックスを利用する1をカウントする回路」を発明の名称とする米国特許第5, 339, 447号。

【0479】1989年11月17日出願され、1993年8月24日に発行された「SIMDモードで作動する際にデータメモリとしてMIMDインストラクションメモリを再利用するデュアルモードのSIMD/MIMDプロセッサ」を発明の名称とする米国特許第5, 239, 654号。

【0480】1989年11月17日出願され現在放棄されている米国特許出願第437, 854号の継続出願である、1992年6月29日に提出された「イメージングコンピュータおよびそのオペレーション方法」を発明の名称とする米国特許出願第07/911, 562号。

【0481】1989年11月17日出願され、1993年7月6日に発行された「集積クロスポイントロジックを有するスイッチマトリックスおよびそのオペレーション方法」を発明の名称とする米国特許第5, 226, 125号。

【0482】1993年11月30日出願された「パレルローテータを備えた3入力端論理ユニット」を発明の名称とする米国特許出願第08/160, 299号。

【0483】1993年11月30日出願された「複数の独立部分および各部分からの結果の表示ビットを記憶するレジスタを有する算術論理ユニット」を発明の名称とする米国特許出願第08/158, 742号。

【0484】1993年11月30日出願された「レジスタペア条件からのメモリストア」を発明の名称とする米国特許出願第08/160, 118号。

【0485】1993年11月30日出願され現在放棄されている米国特許出願第08/160, 115号の継続出願である、1994年10月17日出願された「繰り返しごと複数の商ビットを形成する繰り返し割り算装置、システムおよび方法」。

【0486】1993年11月30日出願された「混合された算術およびブール組み合わせを形成する3入力端算術論理ユニット」を発明の名称とする米国特許出願第08/159, 285号。

【0487】1993年11月30日出願された「単一データワードの複数の等しい部分におけるデータの合計を計算する方法、装置およびシステム」を発明の名称とする米国特許出願第08/160, 119号。

【0488】1993年11月30日出願された「サイズ検出のため最小位ビットの変化を用いるハフマンコード化方法、回路およびシステム」を発明の名称とする米国特許出願第08/159, 359号。

【0489】1993年11月30日出願された「負の数を変換するため条件付き減算を利用するハフマンコード化方法、回路およびシステム」を発明の名称とする米国特許出願第08/160, 296号。

【0490】1993年11月30日出願された「複数の絶対値の差を合計するための方法、装置およびシステム」を発明の名称とする米国特許出願第08/160, 112号。

【0491】1993年11月30日出願された「排他的ORによる最も左側の1の検出を利用する繰り返し割り算装置、システムおよび方法」を発明の名称とする米国特許出願第08/160, 120号。

【0492】1993年11月30日出願された「2つの独立アドレスの選択的マージを利用するアドレス発生器」を発明の名称とする米国特許出願第08/160, 114号。

【0493】1993年11月30日出願された「相関化方法、装置およびシステム」を発明の名称とする米国特許出願第08/160, 116号。

【0494】1993年11月30日出願された「複数の独立したプロセッサオペレーションを制御する長いインストラクションワード」を発明の名称とする米国特許出願第08/160, 297号。

【0495】1993年11月30日出願された「直交データ変換のための回転レジスタ」を発明の名称とする米国特許出願第08/159, 346号。

【0496】1993年11月30日出願された「メディアフィルタ方法、回路およびシステム」を発明の名称とする米国特許出願第08/159, 652号。

【0497】1993年11月30日出願された「条件レジスタソース選択を備えた算術論理ユニット」を発明の名称とする米国特許出願第08/159, 344号。

【0498】1993年11月30日出願された「繰り返しによる除算のための装置、システムおよび方法」を発明の名称とする米国特許出願第08/160, 301号。

【0499】1993年11月30日出願された「冗長コード化された乗算の結果を利用する乗算丸め方法」を発明の名称とする米国特許出願第08/159, 650号。

【0500】1993年11月30日出願された「スプリット乗算方法」を発明の名称とする米国特許出願第08/159, 349号。

【0501】1993年11月30日出願された「ゼロの条件テストを含む混合形条件テストおよびブランチ演算」を発明の名称とする米国特許出願第08/158, 741号。

【0502】1993年11月30日出願された「バックされたワード対の乗算方法」を発明の名称とする米国特許出願第08/160, 302号。

【0503】1993年11月30日出願された「シフトを備えた3入力端算術論理ユニット」を発明の名称とする米国特許出願第08/160,573号。

【0504】1993年11月30日出願された「マスク発生器を備えた3入力端算術論理ユニット」を発明の名称とする米国特許出願第08/159,282号。

【0505】1993年11月30日出願された「パレルローテータおよびマスク発生器を備えた3入力端算術論理ユニット」を発明の名称とする米国特許出願第08/160,111号。

【0506】1993年11月30日出願された「シフトおよびマスク発生器を備えた3入力端算術論理ユニット」を発明の名称とする米国特許出願第08/160,298号。

【0507】1993年11月30日出願された「第2入力と第3入力のブール組み合わせプラス第2入力と第3入力の第2ブール組み合わせが加算された第1入力の合計を計算する3入力端算術論理ユニット」を発明の名称とする米国特許出願第08/159,345号。

【0508】1993年11月30日出願された「第1、第2および第3入力のブール組み合わせプラス第1、第2、第3入力の第2ブール組み合わせの合計を計算する3入力端算術論理ユニット」を発明の名称とする米国特許出願第08/160,113号。

【0509】1993年11月30日出願された「桁上げ伝搬ロジックを利用する3入力端算術論理ユニット」を発明の名称とする米国特許出願第08/159,640号。

【0510】1993年11月30日出願された「書き込み優先権を利用するIF、THENオペレーションのためのデータ処理装置、システムおよび方法」を発明の名称とする米国特許出願第08/160,300号。

【0511】1993年3月8日出願された「MPベクトルインストラクションFP+LOAD/STORE」を発明の名称とする米国特許出願第08/207,989号(TI整理番号15521)。

【0512】1993年3月8日出願された「不動点少数のための正規化方法」を発明の名称とする米国特許出願第08/160,300号(TI整理番号18695)。

【0513】本願は下記の米国特許出願とも関連しており、これらのいずれも参考例としてここに援用する。

【0514】1993年3月8日出願された「PP転送プロセッサにおけるトランスペアレンシーおよび平面マスキング」を発明の名称とする米国特許出願第08/208,413号。

【0515】1993年3月8日出願された「トランスペアレンシーを備えたPIXBLT」を発明の名称とする米国特許出願第08/208,161号。

【0516】1993年3月8日出願された「プロセッ

サからのメッセージパッシングおよびブラストインタラプト」を発明の名称とする米国特許出願第08/208,171号。

【0517】1993年3月8日出願された「X、Y次元を備えたガイド転送および変数ステッピング」を発明の名称とする米国特許出願第08/209,123号。

【0518】1993年3月8日出願された「ガイド転送ラインドロ잉」を発明の名称とする米国特許出願第08/209,124号。

【0519】1993年3月8日出願された「異なるメモリアイプを同時に制御する転送プロセッサ用メモリアイプインターフェース」を発明の名称とする米国特許出願第08/208,517号。

【0520】1993年3月8日出願された「TP転送プロセッサのアーキテクチャ」を発明の名称とする米国特許出願第08/207,503号。

【図面の簡単な説明】

【図1】本発明の画像処理システムのシステムアーキテクチャを示す図。

【図2】単一集積回路マルチプロセッサのアーキテクチャを示す図。

【図3】デジタル画像/グラフィックプロセッサのインタラプトイネーブルおよびインタラプトフラグレジスタを示す図。

【図4】共通ワードのフィールドを示す図。

【図5】デジタル画像/グラフィックプロセッサの通信レジスタを示す図。

【図6】画像システムバスの優先度およびオペレーション方法を示す図。

【図7】クロスバー優先度およびマルチプロセッサ集積回路内のオペレーション方法を示す図。

【図8】本発明の実施例の転送プロセッサ構造を示す図。

【図9】LASTPAGEレジスタの使用を示す図。

【図10】外部メモリアイプセンサのステート図およびオペレーション方法を示す図。

【図11】ソースマシン構造を示す図。

【図12】バケット転送FIFOバッファ、キャッシュバッファ算術演算およびブロックの例を示す図。

【図13】バッファカウンタ算術演算およびブロックの例を示す図。

【図14】aは小エンディアンバケット転送FIFOバッファの例および方法を示す図。bは大エンディアンバケット転送FIFOバッファの例および方法を示す図。

【図15】プロセッサのパラメータメモリの使用例を示す図。

【図16】リンクされたリストバケット転送方法の一例を示す図。

【図17】次元の定められたバケット転送方法の一例を示す図。

【図18】固定パッチのデルタガイド packets 転送方法の一例を示す図。

【図19】固定パッチのオフセットガイド packets 転送方法の一例を示す図。

【図20】固定パッチのオフセットガイドルックアップテーブル packets 転送方法の一例を示す図。

【図21】小エンディアン可変パッチガイドテーブルフォーマットを示す図。

【図22】大エンディアン可変パッチガイドテーブルフォーマットを示す図。

【図23】可変パッチデルタガイド packets 転送方法の一例を示す図。

【図24】可変パッチオフセットガイド packets 転送方法の一例を示す図。

【図25】ソーストランスペアレンシーパラメータを備えた、次元の定められた packets 転送の一例を示す図。

【図26】フィルパラメータを備えた、次元の定められた packets 転送の一例を示す図。

【図27】次元の定められたソースおよび固定パッチガイド宛て先 packets 転送パラメータの一例を示す図。

【図28】次元の定められたソースおよび可変パッチガイド宛て先 packets 転送パラメータの一例を示す図。

【図29】次元の定められたブロック書き込み packets 転送パラメータの一例を示す図。

【図30】 packets 転送パラメータのエンコーディングを示す図。

【図31】 packets 転送タイマー構造を示す図。

【図32】一時中断された packets 転送パラメータの記憶方法の一例を示す図。 *

* 【図33】一時中断された packets サービスパラメータのエンコーディングを示す図。

【図34】小エンディアンモードでの8×ブロック書き込みビット再マッピングにおけるビット再マッピングの一例を示す図。

【図35】小エンディアンモードでの4×ブロック書き込みビット再マッピングにおけるビット再マッピングの一例を示す図。

【図36】シリアルレジスタ転送方法の一例を示す図。

【図37】トランスペアレンシーオペレーションを示す図。

【図38】トランスペアレンシーおよびバイト書き込みロジック回路の実施例を示す図。

【図39】高品位テレビシステムの一実施例を示す図。

【図40】単一のデジタル画像/グラフィックプロセッサを有するマルチプロセッサ集積回路を含むカラーファクシミリシステムの一実施例を示す図。

【符号の説明】

- 1 ホスト処理システム
- 2 ホスト周辺デバイス
- 3 撮像デバイス
- 4 画像キャプチャコントローラ
- 5、6 ビデオラム
- 7 ビデオバレット
- 8 ビデオディスプレイ
- 9 メモリ
- 16 トランシーバ
- 100 マルチプロセッサ集積回路

フロントページの続き

(51)Int. Cl.⁶

G 0 6 F 15/163

識別記号

庁内整理番号

F I

技術表示箇所

G 0 6 F 15/64

4 5 0 C

(72)発明者 キース ブラマー
イギリス国ベッドフォード, サルコム ク
ローズ 6

(72)発明者 クリストファー ジェイ. リード
アメリカ合衆国テキサス州ヒューストン,
パーリングイム 11807

(72)発明者 イェイン ロバートソン
イギリス国ベッドフォードシャー, グレン
ジ レーン, 10

(72)発明者 ニコラス イング - シモンズ
イギリス国ハンチングドン, ケンブリッジ
シャー カウンティ, アルコンベリィ ウ
エストン, ハイフィールド アベニュー,
47

【公報種別】特許法第17条の2の規定による補正の掲載
 【部門区分】第6部門第3区分
 【発行日】平成14年6月7日(2002.6.7)

【公開番号】特開平8-77347
 【公開日】平成8年3月22日(1996.3.22)
 【年通号数】公開特許公報8-774
 【出願番号】特願平7-85856
 【国際特許分類第7版】

G06T 1/60
 G06F 12/00 580
 12/02 560
 15/163

【F I】

G06F 15/64 450 G
 12/00 580
 12/02 560 B
 15/16 320 G
 15/64 450 C

【手続補正書】

【提出日】平成14年3月7日(2002.3.7)

【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】特許請求の範囲

【補正方法】変更

【補正内容】

【特許請求の範囲】

【請求項1】 データ処理装置であって、
 複数の対応するアドレスにデータを記憶するメモリと、
 パケット転送リクエストとパケット転送パラメータとを
 受ける動作をする制御回路であって、前記パケット転送
 パラメータは開始アドレスとアドレスのブロックを定義
 するディメンジョン値とある数のガイドテーブルエン
 トリとテーブルポインタとを含む前記制御回路と、
 前記数のガイドテーブルエントリを有するガイドテー
 ブルであって、各ガイドテーブルエントリはアドレス値を
 含み、前記テーブルポインタは初期状態として前記ガイ
 ドテーブル内の最初のガイドテーブルエントリをポイン
 トする前記ガイドテーブルと、
 前記制御回路と前記ガイドテーブルとに結合されたアド
 レス発生回路であって、前記アドレス発生回路は前記パ
 ケット転送パラメータから、前記開始アドレスと前記デ
 イメンジョン値とガイドテーブルエントリの前記数と前
 記テーブルポインタとを受ける動作をし、前記アドレス
 発生回路は、
 前記開始アドレスと前記テーブルポインタによってポ
 イントされる前記ガイドテーブルエントリの前記アドレ
 ス値の所定の結合により、最初のブロック開始アドレスが
 前記開始アドレスである、ブロック開始アドレスを形成

し、

前記ブロック開始アドレスと前記ディメンジョン値とから
 アドレスのブロックを形成し、アドレスの前記ブロッ
 クにメモリアクセスを行い、
 メモリアクセスが前記ガイドテーブルエントリの数より
 も少ない数しか実行されていなければ、前記メモリアク
 セスに続いて、前記ガイドテーブルの次のエントリをポ
 イントするように前記テーブルポインタを更新すること
 により、

前記ガイドテーブルエントリに対応するメモリアクセス
 のため一組のアドレスのブロックを形成する前記アドレ
 ス発生回路と、を含むデータ処理装置。

【請求項2】 請求項1記載のデータ処理装置であつて、

前記アドレス発生回路が、前記アドレス値を前のブロッ
 ク開始アドレスに加算することにより、前記開始アドレ
 スと前記テーブルポインタによりポイントされた前記ガイ
 ドテーブルエントリの前記アドレス値との前記所定の
 組合せを形成するデータ処理装置。

【請求項3】 請求項1記載のデータ処理装置であつて、

前記アドレス発生回路が、前記テーブルポインタにより
 ポイントされた前記ガイドテーブル値を前記開始アドレ
 スに加算することにより、前記開始アドレスと前記テー
 ブルポインタによりポイントされた前記ガイドテーブル
 エントリの前記アドレス値との前記所定の組合せを形成
 するデータ処理装置。

【請求項4】 請求項1記載のデータ処理装置であつて、

前記パケット転送パラメータの前記ディメンジョン値が、画素のレイの水平ディメンジョン値および垂直ディメンジョン値を含むデータ処理装置。

【請求項5】 請求項1記載のデータ処理装置であって、
前記メモリアクセスが、アドレスの前記ブロックからのメモリ読み出しから成るデータ処理装置。

【請求項6】 請求項1記載のデータ処理装置であって、
前記メモリアクセスが、アドレスの前記ブロックへのメモリ書き込みから成るデータ処理装置。

【請求項7】 請求項1記載のデータ処理装置であって、
更に中央処理ユニットを有し、
前記記憶メモリ、前記制御回路、前記ガイドテーブル、前記アドレス発生回路および中央処理ユニットが単一の集積回路に配置されているデータ処理装置。

【請求項8】 請求項7記載のデータ処理装置であって、更に、
前記単一の半導体チップには含まれない外部メモリにアクセスする、前記単一の集積回路に配置された外部メモリインターフェースを有し、
前記アドレス発生回路は前記メモリと前記外部メモリの両方を含むアドレスを発生するデータ処理装置。

【請求項9】 請求項8記載のデータ処理装置であって、
パケット転送パラメータは前記メモリ内のアドレスの所定の組に記憶され、前記パケット転送パラメータは、ある数のガイドテーブルエン트리と、前記内部メモリのアドレスを指示するイニシャルテーブルポインタの指標とを含み、
前記ガイドテーブルは、前記パケット転送パラメータ内で指示された前記数のガイドテーブルエントリーを有し、前記内部メモリに格納された各ガイドテーブルエントリーはイニシャルアドレスから始まり、ソース開始アドレスとソースアドレス値とアドレスのソースブロックを定義するソースディメンジョン値、および、デスティネーション開始アドレスとデスティネーションアドレス値とアドレスのデスティネーションブロックを定義するデスティネーションディメンジョン値を有し、各ガイドテーブルエントリーは次のガイドテーブルエントリーのイニシャルアドレスを指示するテーブルポインタを有

し、最初のガイドテーブルエントリーは前記パケット転送パラメータの前記テーブルポインタに対応するイニシャルアドレスを有し、
前記アドレス発生回路は、
前記ソース開始アドレスと前記テーブルポインタにより指示された前記ガイドテーブルエントリーの前記ソースアドレス値との所定の結合により、最初のソースブロック開始アドレスが前記開始アドレスである、ソースブロック開始アドレスを形成し、
前記ソースブロック開始アドレスと、前記テーブルポインタにより指示された前記ガイドテーブルエントリーの前記ソースディメンジョン値とからソースアドレスのブロックを形成し、
前記デスティネーション開始アドレスと、前記テーブルポインタにより指示された前記ガイドテーブルエントリーの前記デスティネーションアドレス値との所定の結合により、最初のデスティネーションブロック開始アドレスが前記開始アドレスである、デスティネーションブロック開始アドレスを形成し、
前記デスティネーションブロック開始アドレスと、前記テーブルポインタにより指示された前記ガイドテーブルエントリーの前記デスティネーションディメンジョン値とからデスティネーションアドレスのブロックを形成し、
前記ソースアドレスのブロックから前記デスティネーションアドレスのブロックへデータのメモリ転送を実行し、
実行されたメモリアクセスがガイドテーブルエントリーのの前記数より少ない時、前記テーブルポインタを、前記メモリアクセスに続く前記ガイドテーブル内の次のエントリーを指示する前記現在のガイドテーブルエントリーのの前記テーブルポインタに置き換えることにより、
前記ガイドテーブルエントリーに対応するメモリアクセスのためアドレスのソースおよびデスティネーションブロックの一组を形成し、
前記アドレス発生回路は前記内部メモリの内部アドレスのサブセット内にあるアドレスと、外部メモリに対応する内部アドレスの前記サブセット内に無いアドレスの両方を発生し、前記アドレス発生回路は前記外部メモリインターフェースを介して前記外部メモリにアクセスするデータ処理装置。